Méthodes et outils logiciels pour la conception de sondes oligonucléotidiques pour puces à ADN. Applications aux biopuces transcriptomiques et aux biopuces de type phylogénétique

Sébastien Rimour

N° d'ordre : D. U. : 1691
EDSPIC : 357

UNIVERSITE BLAISE PASCAL – CLERMONT II

Ecole Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse

Présentée par

Sébastien RIMOUR

Pour obtenir le grade de

DOCTEUR D'UNIVERSITE

SPECIALITE : INFORMATIQUE

Méthodes et outils logiciels pour la conception de sondes oligonucléotidiques pour puces à ADN. Application aux biopuces transcriptomiques et aux biopuces de type phylogénétique.

Soutenue publiquement le 6 novembre 2006 devant le jury :

M. Vincent BARRA	Président
M. Gilles BERNOT	Rapporteur et examinateur
M. Hubert CHARLES	Rapporteur et examinateur
M. Pascal SIMONET	Examinateur
M. Pierre PEYRET	Co-directeur de thèse
M. David HILL	Directeur de thèse

Remerciements

Tout d'abord, je tiens à remercier le Professeur Alain Quilliot, directeur du LIMOS et de l'ISIMA, de m'avoir accueilli dans son laboratoire et permis d'effectuer cette recherche dans les meilleures conditions.

J'exprime ma sincère gratitude au Professeur David Hill, mon directeur de thèse, pour son encadrement. Je le remercie pour la confiance qu'il m'a accordée ainsi que pour la liberté qu'il m'a laissée dans le choix de mes projets de recherche.

Je tiens également à remercier le Professeur Pierre Peyret, du Laboratoire de Biologie des Protistes, pour son aide précieuse sur les aspects biologiques de cette thèse et tout le temps qu'il m'a accordé.

J'adresse tous mes remerciements au Professeur Gilles Bernot, de l'université d'Evry, et à M. Hubert Charles, Maître de Conférences à l'INSA de Lyon, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.

Je remercie également M. Pascal Simonet, directeur de recherche au CNRS, de l'intérêt qu'il porte à ce travail en acceptant de participer à mon jury de thèse, ainsi que le Professeur Vincent Barra pour avoir accepté d'en être le président.

Je suis convaincu de l'importance des problématiques biologiques concrètes dans la recherche en bioinformatique, c'est pourquoi j'adresse mes plus vifs remerciements à tous les membres de l'équipe Génomique Intégrée des Interactions Microbiennes, du Laboratoire de Biologie des Protistes. Leur aide m'a été précieuse et les collaborations toujours enrichissantes.

Je tiens également à remercier les personnels enseignants du CUST et de l'IUT de Clermont-Ferrand que j'ai côtoyés au cours de ces années pour leurs conseils et la bonne ambiance de travail qu'ils ont su instaurer.

Je n'oublie pas non plus les membres du LIMOS et de l'ISIMA qui par leur sympathie et leurs bons conseils ont contribué à cette thèse. Plus particulièrement, je remercie les collègues, thésards, post-docs ou stagiaires qui ont partagé mon bureau.

Enfin, j'adresse un grand merci à mes proches, famille et amis, pour leur soutien tout au long de ces années, même dans les moments difficiles.

Merci à Nathalie pour son soutien sans faille et sa patience.

Table des matières

Table des figures

Liste des tableaux

Introduction

Contexte scientifique

J'ai effectué ma thèse au sein du Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS, UMR CNRS 6158) sous la direction du professeur David Hill. Ce laboratoire développe depuis de très nombreuses années une thématique de recherche axée sur la modélisation des systèmes, avec des projets applicatifs, notamment dans le domaine de la biologie. Plus récemment, à la fin des années 90, un petit groupe s'est constitué au sein du laboratoire autour d'une discipline émergente : la bioinformatique.

Plus précisément, le travail porte sur l'informatique des puces à ADN, ou biopuces. Cette technologie de biologie moléculaire se révèle très puissante pour identifier et quantifier les constituants d'un mélange d'ADN ou d'ARN grâce à l'hybridation en parallèle de plusieurs milliers de sondes nucléiques greffées sur un support miniature. Le nombre de données manipulées lors d'expériences de puces à ADN est tellement important que l'on ne peut pas envisager à l'heure actuelle de les interpréter dans toute leur complexité sans outils informatiques. Bien que relativement récente, l'utilisation de cette technique connaît un succès croissant. La puce à ADN peut être considérée simplement comme un outil de biologie moléculaire, et dans ce cas, l'utilisateur biologiste cherche uniquement à interpréter les données qu'elles produisent. Des sociétés spécialisées commercialisent ces puces pour les laboratoires de recherche ou l'industrie pharmaceutique. Néanmoins, les biopuces peuvent également faire l'objet de travaux de recherche, car leur concept repose sur un procédé extrêmement complexe et multidisciplinaire faisant intervenir la biologie, la chimie, les nanotechnologies et l'informatique. Rien que dans cette dernière discipline, une expérience de puces à ADN fait intervenir des domaines aussi différents que l'algorithmique, les systèmes d'informations, l'imagerie, ou encore les statistiques et l'analyse de données. C'est pourquoi de nombreuses recherches sont menées sur les concepts fondamentaux qui constituent une expérience de biopuce. Au sein du LIMOS, de nouvelles méthodes ont été développées dans le domaine de l'analyse des images et de l'analyse des données.

La problématique de ma thèse est née d'une collaboration avec l'équipe Génomique Intégrée des Interactions Microbiennes (GIIM) dirigée par le professeur Peyret du Laboratoire de Biologie des Protistes (LBP, UMR CNRS 6023). Cette équipe mène entre autre des études de génomique sur un parasite eucaryote intracellulaire obligatoire : *Encephalitozoon cuniculi*. Une collaboration avec le Genoscope a permis de séquencer et d'analyser la totalité du génome de ce parasite. Des études de transcriptomique permettent maintenant de décrypter la physiologie de ce pathogène pouvant servir de modèle pour d'autres parasites eucaryotes. Elles permettent aussi de mieux cerner les mécanismes de régulation de l'expression des gènes d'un eucaryote à génome réduit. Les thématiques de recherche développées par l'équipe GIIM portent également sur la caractérisation des communautés microbiennes de l'environnement et sur l'identification des voies métaboliques d'intérêt biotechnologique ou impliquées dans les processus de bioremédiation. La bioremédiation consiste à éliminer toutes formes de pollution en utilisant les capacités métaboliques naturelles des microorganismes. Là encore, une approche puce à ADN a été retenue afin de caractériser les micro-organismes de l'environnement. Ainsi, l'équipe GIIM a développé une collaboration avec le LIMOS autour des problématiques bioinformatiques qui découlent de telles expériences.

Une approche interdisciplinaire

La bioinformatique est un domaine interdisciplinaire par nature. Elle propose (entre autre) des méthodes et des outils informatiques qui permettent de gérer et d'analyser l'information génétique. Certains en font une branche de la biologie, d'autres préfèrent la définir comme une discipline à part entière. Elle fait appel à des connaissances en biologie, en informatique, mais aussi en mathématique. Avec l'apparition de technologies de biologie moléculaire de plus en plus complexes, on voit aujourd'hui émerger des projets pluridisciplinaires où collaborent biologistes et informaticiens, et souvent aussi chimistes, physiciens, mathématiciens, etc. Certains informaticiens choisissent de se concentrer sur des approches purement théoriques en apportant des méthodes et algorithmes issus de leur domaine et en les appliquant aux données génétiques. De même que certains biologistes souhaitent seulement utiliser l'outil informatique pour répondre à leurs questions sans se soucier des théories sous-jacentes. Cependant, il me semble également important que ces projets pluridisciplinaires de bioinformatique incluent des scientifiques ayant des connaissances dans les deux domaines, biologie et informatique, et capables de communiquer avec des chercheurs des deux secteurs.

Informaticien de formation, et touchant du doigt le domaine passionnant de la biologie moléculaire et de la génétique, j'ai souhaité adopter au cours de ma thèse une approche totalement interdisciplinaire, en essayant d'acquérir des connaissances en biologie afin de résoudre au mieux les problèmes posés. Cette volonté m'a notamment amené à suivre des cours de biologie moléculaire au début de mes travaux. Ce mémoire a donc la modeste ambition de présenter une vision « bio-informatique » des recherches effectuées, et non purement informatique, comme on pourrait le supposer au regard de ma section de rattachement.

Problématique

L'équipe Génomique Intégrée des Interactions Microbiennes souhaitait réaliser des expériences de puces à ADN, utilisant des sondes oligonucléotidiques, pour répondre à deux problématiques distinctes. D'une part elle désirait mener des études transcriptomiques sur son modèle d'étude : le pathogène *Encephalitozoon cuniculi*. D'autre part, elle voulait suivre l'évolution des communautés bactérienne du sol lors d'un procédé de bioremédiation. Elle a développé une coopération avec le LIMOS afin d'approfondir les aspects informatiques de ces expériences et de ne pas se concentrer uniquement sur les aspects biologiques. En effet, l'équipe GIIM ne souhaitait pas simplement utiliser les puces à ADN comme outil technologique mais également développer en collaboration avec le LIMOS de nouvelles méthodes applicables à n'importe quelle expérience de ce type.

Une des problématiques qui s'est dégagée de cette collaboration entre le LIMOS et l'équipe GIIM et qui fait l'objet de cette thèse concerne l'étape de conception d'une expérience de biopuces. En effet, l'utilisation de puces à oligonucléotides nécessite une étape de conception des sondes : ces courtes séquences de quelques dizaines de bases devront constituer des étiquettes spécifiques des gènes étudiés, et devront en outre vérifier un certain nombre de critères biologiques. La détermination des sondes à déposer sur une puce à ADN présente de nombreuses difficultés, à la fois algorithmiques et biologiques. Nous verrons également que les particularités des études menées par l'équipe GIIM ajoutent des problèmes supplémentaires par rapport aux expériences classiques. L'objet de cette thèse est donc le développement d'algorithmes et d'outils logiciels pour la conception de sondes

oligonucléotidiques dans le cadre d'expériences de puces à ADN. Ces outils doivent permettre aux biologiste du LBP de concevoir des expériences pour deux problématiques distinctes : d'une part l'étude globale de l'expression des gènes d'un parasite eucaryote intracellulaire obligatoire au cours de son cycle de développement, et d'autre part la conception de biopuces dites phylogénétiques pour l'identification des micro-organismes présents dans un environnement complexe.

Organisation du mémoire

Ce mémoire est divisé en six chapitres et une annexe. Les chapitres I, II et III portent sur la synthèse bibliographique alors que les chapitres IV, V et VI présentent mes travaux de recherche.

Le **chapitre I** présente le contexte général de la thèse. Il décrit l'ensemble des étapes, biologiques et informatiques, d'une expérience de puces à ADN. Il présente également la problématique biologique ayant guidé les recherches de cette thèse.

Le **chapitre II** dresse un état de l'art de la conception d'oligonucléotides pour puces à ADN. Il présente les méthodes et algorithmes utilisés, ainsi que les logiciels existants.

Le **chapitre III** s'intéresse aux expériences de biopuces en général, mais cette fois vues sous l'angle du Génie Logiciel. Nous tentons ici de rechercher des solutions aux problèmes de réutilisabilité des composants logiciels soulevés aux chapitre II. Après avoir présenté les notions de Génie Logiciel nécessaires à la compréhension de la suite (« Model Driven Architecture », ontologies), nous dressons un bilan de l'utilisation des techniques récentes de ce domaine pour les puces à ADN.

Le **chapitre IV** propose des solutions aux problèmes rencontrés dans la conception de sondes pour puces à ADN. Il donne d'une part des solutions sur le plan méthodologique et algorithmique, et d'autre part des solutions sur le plan du Génie Logiciel avec un modèle orienté-objet proche de la philosophie MDA.

Le **chapitre V** présente une application des méthodes et algorithmes exposés au chapitre IV : le développement d'un logiciel de conception de sondes pour puces à ADN. Il détaille également l'utilisation de ce logiciel pour la conception d'une biopuce « transcriptome » complète du parasite *Encephalitozoon cuniculi*. La base de donnée et l'interface Web résultant de cette conception sont décrites.

Le **chapitre VI** décrit une seconde application réalisée à partir des connaissances acquises sur la conception de sondes. Il propose un algorithme de détermination d'oligonucléotides pour des puces de type phylogénétique, destinées à l'identification d'organisme. Ce chapitre décrit également la parallélisation de cet algorithme ainsi que son déploiement sur une architecture de type cluster de calcul.

Enfin, l'**annexe** constitue une introduction à la biologie moléculaire et à la génétique. Elle a été rédigée à destination des informaticiens non familiers de ce domaine, afin d'accentuer le caractère interdisciplinaire de cette thèse.

Chapitre I

Contexte et problématique biologique

1 La technologie des puces à ADN

Le terme puce à ADN (ou « DNA microarray » en anglais) désigne une technologie miniaturisée de biologie moléculaire permettant l'analyse d'échantillons biologiques complexes. Son concept repose sur un procédé multidisciplinaire intégrant la biologie, la chimie des acides nucléiques, les nanotechnologies, l'analyse d'images et la bioinformatique. Une puce à ADN est constituée d'un support solide, généralement une lame de verre, sur lequel sont fixés de manière ordonnée des milliers de fragments d'ADN appelées sondes. Ces sondes seront caractéristiques d'un gène donné ou d'une région d'ADN, et constitueront des « étiquettes » ou « codes barres » spécifiques. La position de ces fragments sur la biopuce est connue. Ainsi, par mise en contact de ce support avec un mélange complexe marqué par fluorescence, il est possible de détecter et de quantifier l'ensemble des cibles que contient ce mélange en une seule expérience.

La technique de détection d'un fragment d'ADN par hybridation avec son brin complémentaire marqué radioactivement, dite technique du « Southern Blot », était connue depuis les années 70 [Southern 1975]. Mais c'est à la fin des années 90 que ce principe a été étendu en fixant des milliers de dépôts d'ADN sur un support solide miniature, permettant ainsi la détection simultanée d'un très grand nombre de séquences en parallèle. Les premières puces à ADN ont été fabriquées par l'équipe de P. Brown à l'université de Stanford [DeRisi et al 1997], et contenaient le génome de la levure du boulanger (*Saccharomyces cerevisae*). La technologie des biopuces connaît donc un développement exponentiel depuis la « révolution génomique » avec la capacité de séquencer des génomes complet. La base de données GOLD (Genome OnLine Database[1]) recense en 2006 plus de 2000 projets de séquençages complets de génomes, parmi lesquels 400 sont achevés et publiés.

Les premières puces à ADN étaient fabriquées en déposant les séquences complètes des gènes (ADNc, produits PCR) sur le support solide et servaient le plus souvent à détecter les transcrits d'une cellule (ARNm). Aujourd'hui, la technique a considérablement évoluée [Barrett 2005], et son utilisation est de plus en plus courante dans les laboratoires de biologie. Une simple recherche du mot « microarray » dans la base de données bibliographique PubMed renvoie près de 13000 articles. De nouveaux types de puces sont apparus, la principale évolution étant l'apparition de supports sur lesquels ne sont plus fixés les séquences complètes des gènes, mais des séquences « courtes » de type oligonucléotidique (50 à 70 bases) conçues pour être spécifiques de chaque gène. D'autre part, les applications se sont diversifiées : elles vont de la classique étude de l'expression transcriptionnelle à l'identification des sites d'interaction de protéines avec l'ADN [Ren et al 2000, Buck and Lieb 2004], la détection du nombre de copies d'ADN [Pinkel et al 1998, Snijders et al 2001], la détection de mutations (SNP Single-Nucleotide Polymorphism) [Hirschhorn et al 2000, Kennedy et al 2003, Tebbutt et al 2004], la comparaison de génomes [Behr et al 1999], l'identification de micro-organismes. Le principe des puces à ADN a même été étendu aux proteines puisqu'il existe maintenant des « protein microarrays » sur lesquelles sont fixées

[1] www.genomesonline.org

non plus des séquences d'ADN mais des protéines. Enfin, en quelques années, le mode de fabrication des puces a évolué : les premières puces étaient la plupart du temps fabriquées « sur-mesure » par le laboratoire voulant réaliser l'expérience. Aujourd'hui, si ce mode de fabrication reste possible, il existe également une forte compétition économique entre des sociétés spécialisées qui commercialisent les puces pour les laboratoires de recherche ou l'industrie pharmaceutique.

La Figure 1 illustre le principe de fonctionnement d'une biopuce classique. Il est important de noter le vocabulaire utilisé : le terme sonde (« probe » en anglais) désigne le fragment d'ADN fixé sur le support solide, alors que le terme cible (« target » en anglais) désigne les séquences nucléiques marquées contenues dans l'échantillon à analyser. De même, on appelle « spot » une unité d'hybridation située sur le support solide, dans laquelle est fixée une micro gouttelette d'une sonde donnée.

2 Les différents types de puces à ADN

Les différents types de puces à ADN peuvent être classés suivant plusieurs critères : selon le type des sondes utilisées, ou selon la méthode de fixation des sondes sur le support solide.

2.1 Types de sondes utilisées

En ce qui concerne les sondes fixées sur le support solide, elles peuvent être de deux types (Figure 2). Elles peuvent être constituées d'ADN préparé par amplification PCR à partir du génome ou de banques d'ADN complémentaires. Ces séquences sont dites « longues » puisqu'elles sont de longueur voisine des séquences qu'elles ciblent. L'avantage de ce type de sondes est qu'il n'est pas nécessaire de connaître les séquences pour faire la puce, et de plus leur sensibilité est très bonne. Cependant leur préparation est laborieuse et le coût de préparation est élevé.

Le second type de sonde utilisé est l'oligonucléotide : il s'agit d'un petit segment d'ADN (quelques dizaines de nucléotides) simple brin synthétisé chimiquement, dont la séquence sera choisie très précisément pour être complémentaire d'une petite partie du gène à cibler. Ce type de sonde a tendance aujourd'hui à supplanter les séquences longues préparées par PCR. En effet, leur préparation est plus simple et elles présentent une bonne sensibilité pour ce qui est des oligos longs (50-70 mers). Néanmoins, leur inconvénient principal est qu'il est nécessaire de connaître les séquences des cibles pour déterminer les sondes. De plus le choix des séquences à utiliser est complexe si l'on souhaite éviter les hybridations croisées avec des séquences non cibles. Nous ne développerons pas plus ce problème ici puisqu'il fait l'objet d'une grande partie de cette thèse.

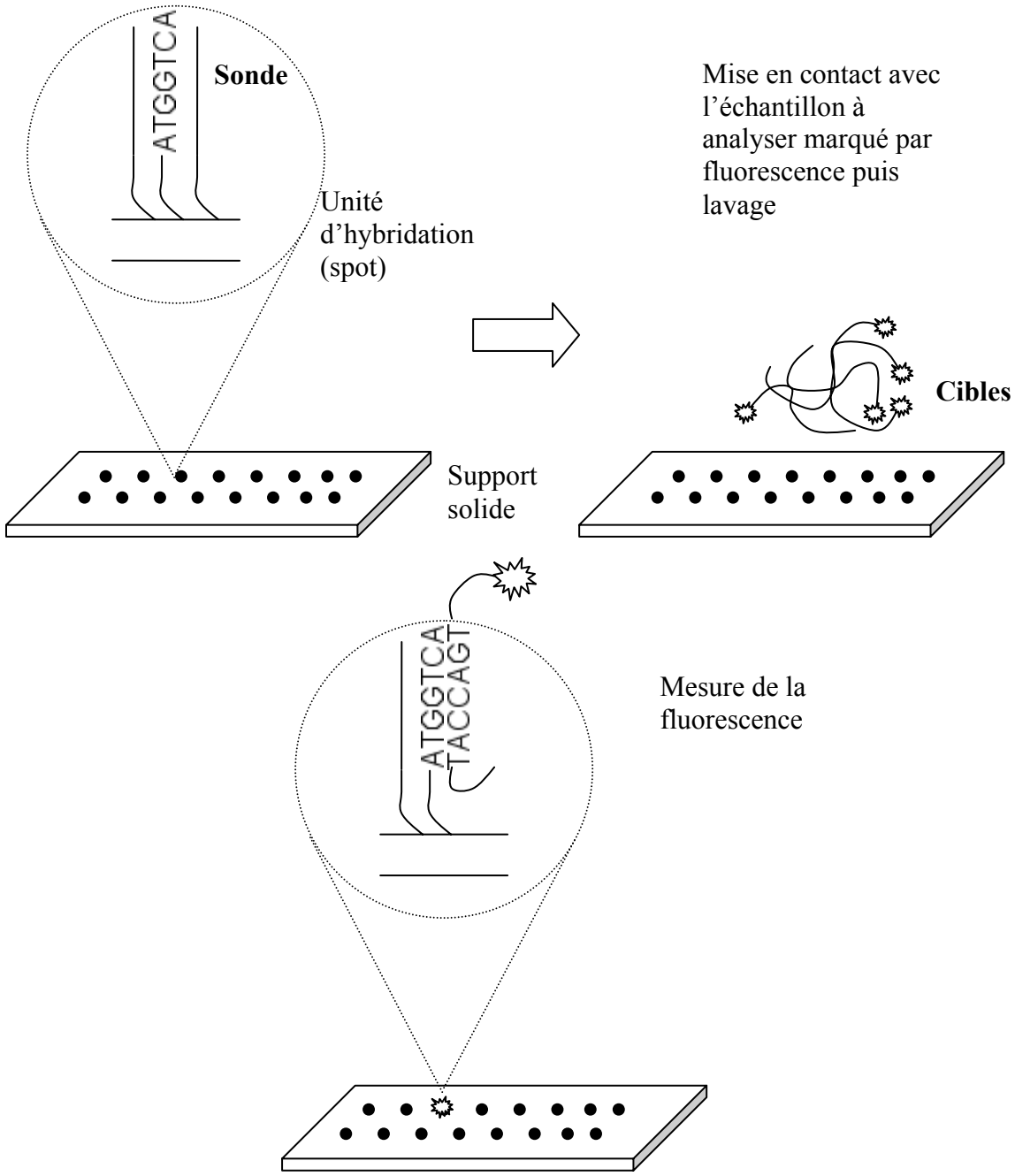


Figure 1 : Principe de fonctionnement d'une puce à ADN.

La puce est généralement constituée d'une lame de verre de quelques centimètres carré sur laquelle sont fixées de manière ordonnée des milliers d'unités d'hybridation (spot). Chaque spot contient une micro gouttelette d'une solution d'ADN de séquence connue (sonde). Après mise en contact avec un échantillon marqué par fluorescence puis lavage, une mesure de la fluorescence de la puce permet d'indiquer si la séquence complémentaire de chaque sonde était présente dans l'échantillon et d'en mesurer l'abondance.

2.2 Méthode de fixation des sondes sur le support solide

Les puces à ADN peuvent également être distinguées par la méthode de fixation des sondes sur le support solide. Cette fixation peut être opérée par un robot spotteur. C'est la technique qui a été utilisée pour fabriquer les premières puces à la fin des années 90. La fabrication de la puce s'effectue en trois étapes distinctes :

- fabrication des sondes à fixer sur le support solide
- dépôt des sondes sur le support par le robot spotteur
- traitement du support afin d'empêcher tout autre brin d'ADN de se fixer au support.

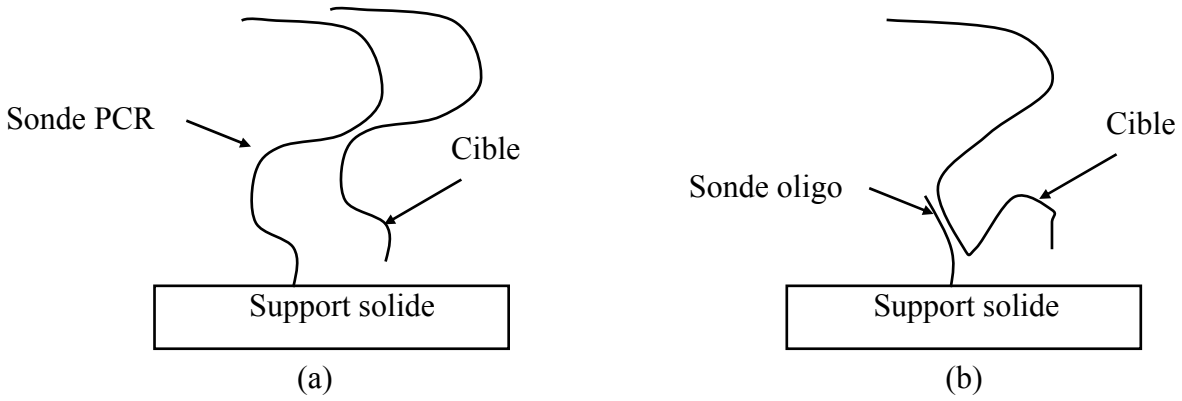


Figure 2 : illustration des deux principaux types de sondes utilisés pour les puces à ADN.

(a) Dans le cas de sondes préparées par amplification PCR, l'hybridation se fait sur toute la longueur de la cible.
(b) Avec une sonde oligonucléotidique, l'hybridation à lieu avec une petite partie de la cible.

La fixation des sondes au support solide se fait par liaison chimique ou électrostatique.

La seconde méthode de fabrication des puces est la synthèse des sondes *in situ* : dans ce cas, les sondes sont obligatoirement des oligonucléotides, et ces derniers sont synthétisés base par base directement sur la puce. Chaque base ajoutée est munie d'un système de protection chimique qui empêche la fixation de toute autre base à sa suite. Grâce à un système de déprotection sélectif (ou photodéprotection), on peut ainsi synthétiser l'ensemble des oligos de la puce. Il existe ensuite différents mécanismes de photodéprotection, souvent brevetés par des sociétés privées :

- la photodéprotection par masque : c'est la base de la technologie Affymetrix. Avec cette technique, également appelée photolithographie, un masque est placé entre une source de lumière et le support solide à chaque étape de la synthèse, permettant ainsi la déprotection d'un ensemble de spots sélectionnés.

- la photodéprotection sans masque : la lumière est dirigée vers les spots sélectionnés grâce à un système de micro miroirs.
- La synthèse type « jet d'encre » : utilisée notamment par Agilent et Rosetta inpharmaceutics. La déprotection est chimique avec un système de projection des bases sur les spots sélectionnés via un système similaire aux imprimantes à jet d'encre.

3 Les étapes d'une expérience de puces à ADN

Afin d'illustrer le principe de fonctionnement d'une puce à ADN, nous détaillons ici les différentes étapes d'une expérience classique : la mesure de l'expression transcriptionnelle des gènes. C'est historiquement la première application des puces à ADN et elle est encore largement utilisée aujourd'hui. Il s'agit de mesurer, pour un ensemble de gène (1 gène = 1 spot sur la puce) l'abondance d'ARN messager produit par chaque gène dans des conditions données. Une mesure du « niveau d'expression » de chaque gène est ainsi réalisée.

Dans un premier temps, la puce est fabriquée avec l'un des procédés décrits au paragraphe 2. Le plus souvent, chaque spot cible un gène, et contient donc une solution d'ADN complémentaire de la séquence du gène. Ensuite, l'expérience est constituée d'une succession d'étapes, biologiques puis informatiques, permettant d'aboutir à la mesure de l'expression transcriptionnelle des gènes.

3.1 Préparation et marquage de l'échantillon cible

La première étape consiste à extraire l'ARN messager des échantillons de culture cellulaire à analyser. En pratique, on extrait les ARN totaux puis on les purifie éventuellement afin de ne conserver que les ARN messagers. Cette étape n'est pas obligatoire car il est possible de travailler directement avec les ARN totaux. Le plus souvent, on ne cherchera pas à mesurer l'abondance des ARNm en valeur absolue car il peut exister des différences de marquage entre les messagers de gènes différents. C'est pourquoi on utilise plutôt deux échantillons, un échantillon témoin et l'échantillon à analyser, et on mesure l'abondance relative des ARNm de ce dernier par rapport à l'échantillon témoin.

Une rétrotranscription est ensuite réalisée en présence de nucléotides modifiés permettant de coupler un marqueur fluorescent. Les deux fluorochromes les plus utilisés sont les carbocyanines Cy3 et Cy5, on parle alors respectivement de fluorochrome vert et de fuorochrome rouge, en référence à la longueur d'onde du laser utilisé pour exciter la molécule. Des cibles d'ADN complémentaires (ADNc) représentatives de l'ensemble des gènes exprimés pour chaque culture sont ainsi obtenues (voir Figure 3).

3.2 Hybridation et lavage

L'étape suivante est la mise en contact des cibles marquées et du support solide : c'est la phase d'hybridation. Si elles sont complémentaires, une molécule cible simple brin et une sonde fixée sur la lame, également simple brin, se lient pour former un complexe d'ADN en double hélice. L'hybridation est un phénomène complexe très dépendant des conditions expérimentales : température, concentration en sels, volume de solution cible, etc.

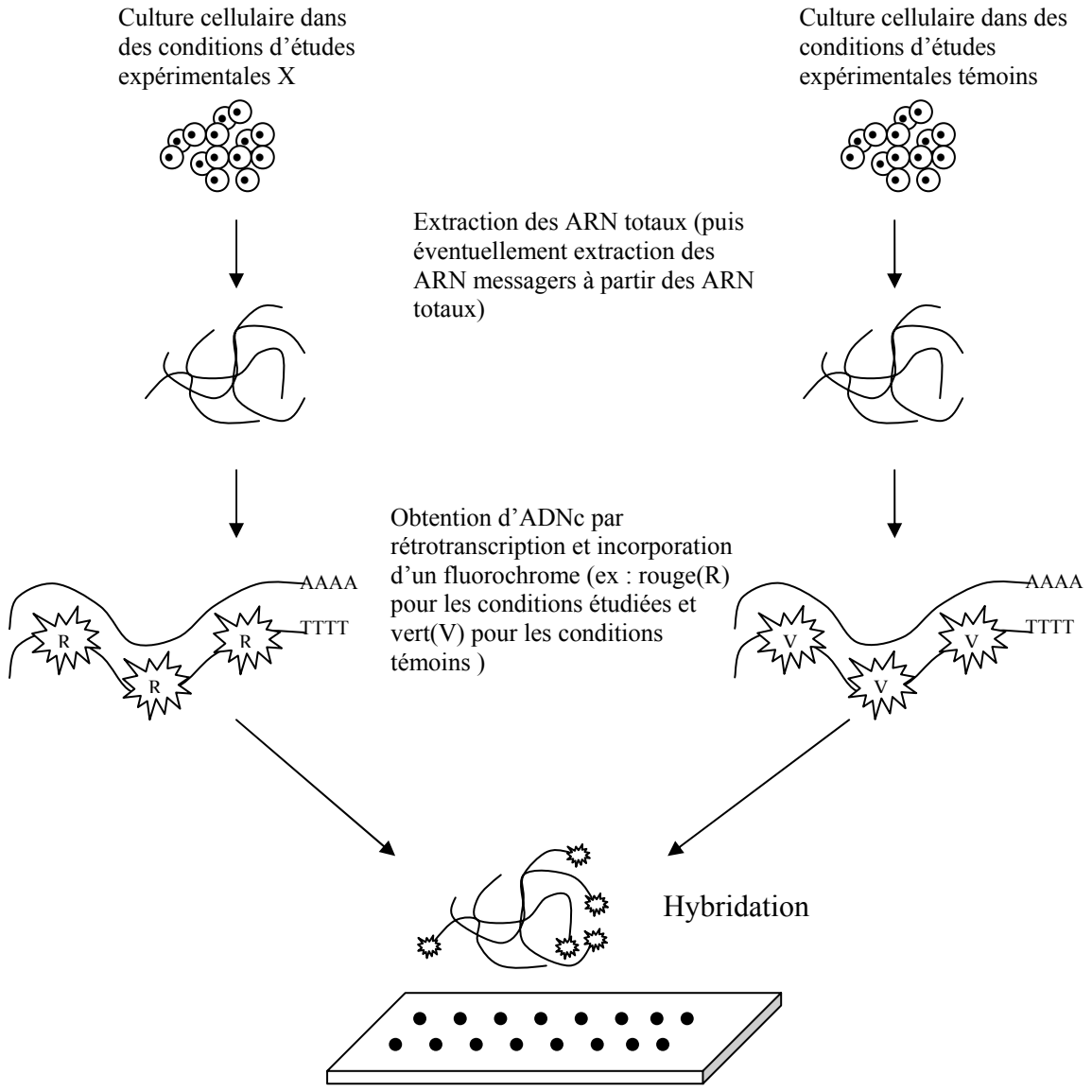


Figure 3 : Principe de la préparation des cibles pour une analyse d'expression transcriptionnelle par puce à ADN.

Il existe deux méthodes d'hybridation : manuelle ou automatique. Dans le cas d'une hybridation manuelle, la lame est placée dans une chambre d'hybridation, puis l'expérimentateur dépose la solution cible au contact des spots de la puce recouverte d'une lamelle. La chambre est ensuite placée dans un incubateur qui maintient la puce à une température donnée. La plupart du temps, cette phase d'hybridation dure entre 12h et 24h. Dans le cas d'une hybridation automatique, tout se passe dans une station d'hybridation, où un robot réalise les opérations de dépôt d'échantillon sur la lame, d'incubation et de lavage, souvent pour plusieurs lames en même temps. L'avantage est alors un meilleur contrôle de la température de la lame et des cibles ainsi qu'une réduction de la variabilité entre les hybridations et les expérimentateurs.

Après la phase d'hybridation, la lame est lavée afin d'enlever l'excès de cibles non hybridées et de s'assurer que seules les cibles marquées qui se sont hybridées avec leurs sondes respectives demeurent sur la puce.

3.3 Acquisition de l'image

La dernière étape expérimentale d'une expérience de puce à ADN est la production d'une image numérique de la surface de la lame. Cette dernière est placée dans un scanner qui va envoyer un signal lumineux de longueur d'onde appropriée sur les spots (Figure 4). Si le spot contient des molécules fluorescentes, donc des cibles hybridées, celles-ci vont être excitées et renvoyer un signal détecté par un photomultiplicateur (PMT).

Dans une expérience classique à deux couleurs, la sortie du scanner est constituée de deux images monochromes : une pour chaque longueur d'onde. La plupart du temps, l'intensité de signal mesurée pour un pixel est quantifiée par un nombre codé sur 16 bits, entre 0 et 65535.

Avec une résolution de 10µm par pixel, une lame classique produira une image de 7500x2200 pixels. Le format de sortie du scanner est généralement le TIFF (Tagged Image File Format).

3.4 Analyse de l'image

L'image numérique produite par le scanner (Figure 5) est analysée par un logiciel d'analyse d'images de puces à ADN. Cette étape est la première du processus d'analyse des données, et a un fort impact sur la qualité des résultats obtenus au final. L'objectif est d'obtenir à partir de l'image brute, pour chaque spot, une mesure de la quantité de cible hybridée. Ceci est réalisé en trois étapes :

- L'adressage : c'est l'identification de la position des spots.
- La segmentation : identifier quels sont les pixels qui appartiennent au spot et quels sont ceux qui appartiennent au fond.
- Le calcul des valeurs numériques : intensité du spot, intensité du fond, et autres valeurs permettant de contrôler la qualité de la mesure.

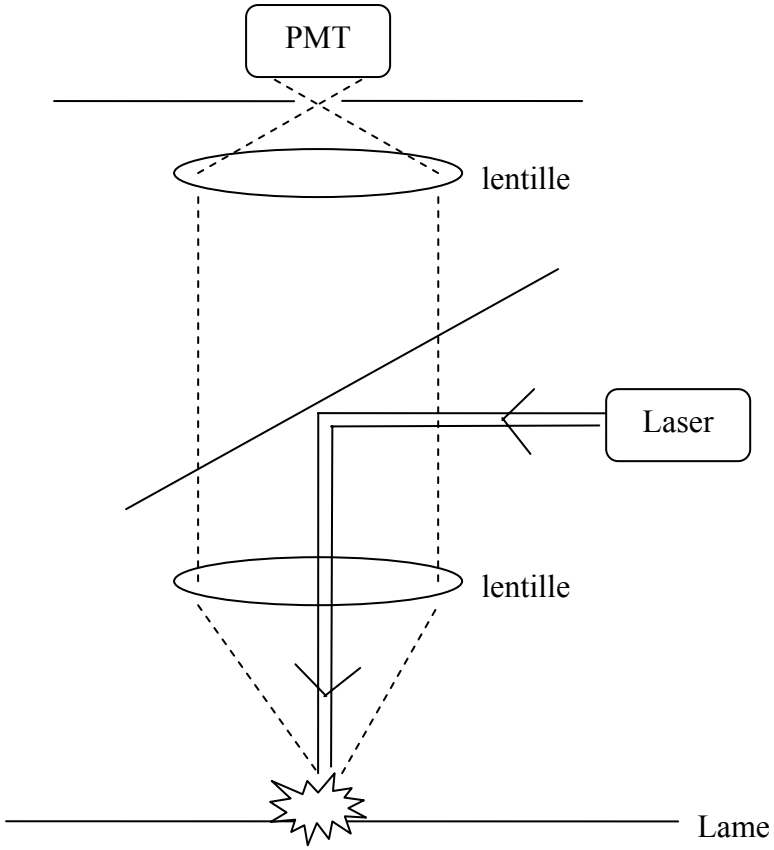


Figure 4 : Principe de fonctionnement d'un scanner de puce à ADN.

Un laser est utilisé pour exciter les molécules fluorescentes incorporées aux cibles. Le signal lumineux émis est alors capté par un photomultiplicateur (PMT) et converti en signal numérique.

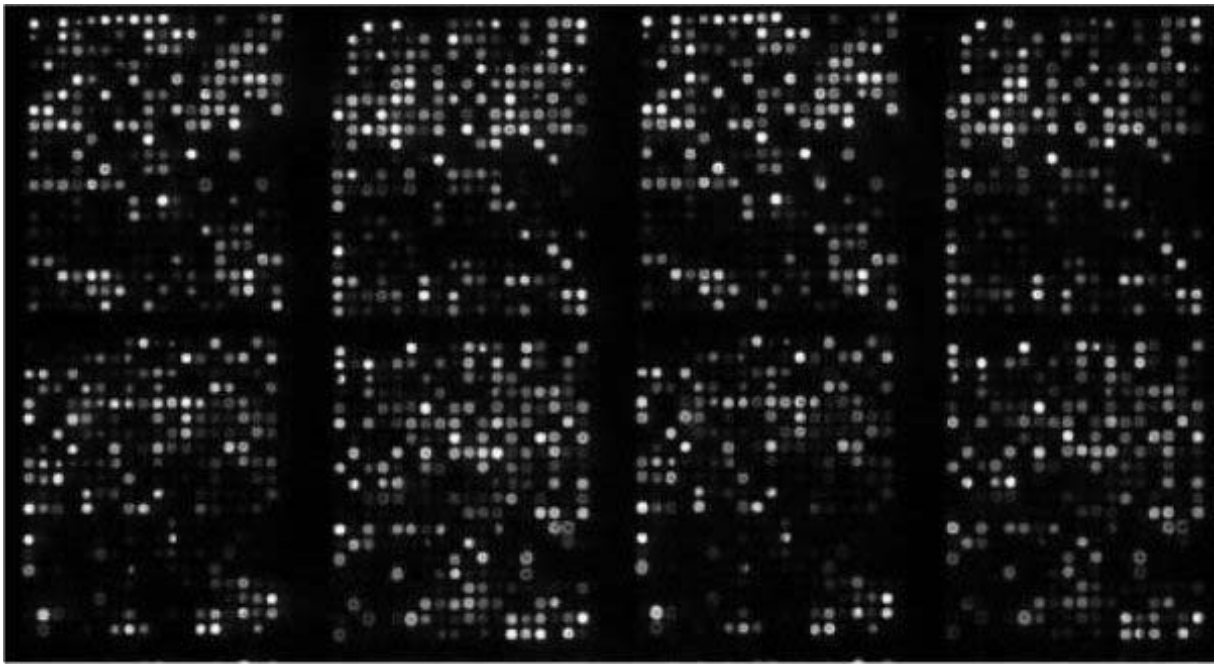


Figure 5 : exemple d'image brute d'une puce à ADN produite par le scanner.

3.4.1 L'adressage

Il s'agit d'identifier la position des spots sur la lame afin de définir pour chaque spot, une région le contenant et contenant uniquement celui-ci. On parle souvent d'appliquer une grille sur l'image, car les régions considérées sont souvent rectangulaires. En effet, les spots sont organisés sur la lame en motifs réguliers : des blocs rectangulaires dans lesquels on trouve un certains nombre de spots espacés régulièrement.

Le positionnement de la grille sur l'image est facilité par le fait que la structure de base est connue : elle est déterminée par le robot spotter, ou le cas échéant par la machine effectuant la synthèse *in situ*. Donc, dans un premier temps, la grille est positionnée approximativement, à partir des paramètres caractéristiques du plan de dépôt : nombre de lignes de blocs, nombre de colonnes de blocs, espacement entre les blocs, nombre de spots par blocs, espacement entre les spots à l'intérieur d'un bloc. Cependant, le processus mécanique de fixation des sondes n'est jamais parfait, et il existe toujours des petites variations dans la position des spots par rapport aux paramètres du plan de dépôt. Il est donc nécessaire d'effectuer un ajustement de la grille en la déformant afin de déterminer correctement les régions contenant chaque spot. Cet ajustement est soit manuel, soit réalisé automatiquement par l'algorithme d'adressage.

3.4.2 La segmentation des spots

Une fois déterminée une région contenant un spot et un seul, il s'agit de déterminer quels sont les pixels qui appartiennent au spot (dont la valeur de l'intensité participera à la mesure globale du signal du spot) et quels sont les pixels appartenant au fond (dont l'intensité participera à la mesure du bruit de fond). Il existe diverses méthodes de segmentation.

La première méthode a avoir été utilisée est la méthode du cercle fixe : un cercle de taille fixe est placé sur chaque spot et tous les pixels situés à l'intérieur de ce cercle sont utilisés pour calculer l'intensité su spot. Cette technique ne produit pas de très bons résultats étant donné que les spots ont rarement un diamètre uniforme sur l'ensemble de la lame. Ce diamètre peut notamment dépendre de la quantité de solution déposée. La plupart des logiciels proposent maintenant une version améliorée de cette approche en adaptant le diamètre du cercle à chaque spot : c'est la méthode dite du cercle adaptatif (Figure 6).

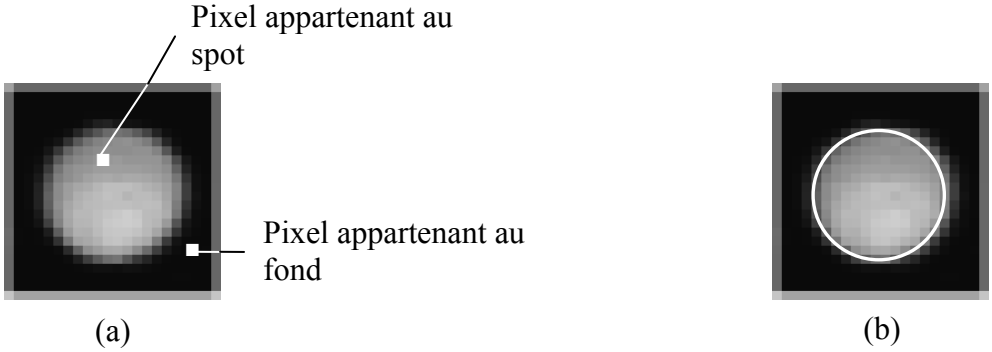


Figure 6 : Segmentation d'un spot sur une image de puce à ADN.

L'objectif est de déterminer quels sont les pixels appartenant au spot et quels sont ceux appartenant au fond (a). Ceci peut être réalisé en utilisant la méthode du cercle adaptatif (b).

La deuxième grande classe de méthode de segmentation concerne tous les algorithmes qui travaillent sur un histogramme des intensités des pixels de la région considérée. Des seuils d'intensité sont définis pour déterminer quels sont les pixels qui appartiennent au spot et quels sont ceux qui appartiennent au fond. Ce type de méthode produit de bons résultats lorsque les spots ont des formes irrégulières (donc non circulaire), mais peut s'avérer instable pour des spots très petits.

Enfin, les algorithmes les plus sophistiqués, dits à contour adaptatif, cherchent à déterminer la forme exacte des spots.

3.4.3 Le calcul des valeurs numériques

Un fois le spot segmenté, le logiciel calcule les valeurs numériques caractéristiques du spot :

- Moyenne et médiane des intensités des pixels du spot.
- Moyenne et médiane des pixels d'une région du fond situé autour du spot.

Le logiciel permet également de repérer visuellement les spots non exploitables (hybridation absente, poussière sur la lame…) en les marquant avec des drapeaux (ou « flags » en anglais), ceci afin de ne pas les prendre en compte lors de l'analyse des données.

Il s'agit ensuite d'extraire une valeur numérique qui sera représentative du niveau d'expression relatif du gène considéré. Il existe plusieurs manières de calculer cette valeur, et les avis divergent sur la pertinence des différentes méthodes. On peut ainsi choisir de considérer ou non la valeur du bruit de fond. Le niveau d'expression relatif du gène i dans l'échantillon marqué en rouge par rapport à l'échantillon marqué en vert peut être calculé par l'une des deux formules suivantes :

$$R_{r/v}(i) = \frac{\text{intensité médiane du spot i en rouge – médiane du bruit de fond local en rouge}}{\text{intensité médiane du spot i en vert – médiane du bruit de fond local en vert}}$$

ou

$$R_{r/v}(i) = \frac{\text{intensité médiane du spot i en rouge}}{\text{intensité médiane du spot i en vert}}$$

Souvent les intensités médianes sont utilisées plutôt que les intensités moyennes, car elles sont moins sensibles au biais provoqués par les valeurs extrêmes. Le ratio est supérieur à 1 si le gène i est plus exprimé dans l'échantillon marqué en rouge que dans l'échantillon marqué en vert et inversement.

Même si les ratios des intensités donnent une représentation intuitive des différences d'expression des gènes, ils ont l'inconvénient de traiter différemment les gènes surexprimés et les gènes sous exprimés. Ainsi, un gène qui s'exprime deux fois plus dans l'échantillon marqué en rouge que dans l'échantillon marqué en vert aura un ratio égal à 2 alors qu'un gène qui s'exprime deux fois moins aura un ratio de 0,5. La transformation la plus courante que l'on applique aux données brutes est donc de considérer les logarithmes ($\log_2$ en général) des ratios, ce qui a l'avantage de produire un spectre continu de valeurs.

Pour un gène qui s'exprime deux fois plus dans l'échantillon marqué en rouge que dans l'échantillon marqué en vert : $\log_2 (2/1) = 1$.

Inversement, pour un gène qui s'exprime deux fois moins dans l'échantillon marqué en rouge que dans l'échantillon marqué en vert : $\log_2 (1/2) = -1$.
Enfin, si le gène s'exprime de la même façon dans les deux échantillons : $\log_2(1/1) = 0$.

3.5 Normalisation des données

Du fait de la complexité de mise en œuvre d'une expérience de puce à ADN, il existe de nombreux facteurs pouvant faire varier le signal fluorescent mesuré. Il est donc difficile de mettre directement en relation les intensités mesurées lors de l'analyse d'image et les niveaux d'expression des gènes. Il est donc nécessaire de passer par une étape de normalisation des données, afin de distinguer les variations biologiques et expérimentales (celles que l'on veut mettre en évidence) des variations systématiques. Ces dernières ont pour origine en particulier :

- les différences dans les rendements de marquage par les fluorochromes Cy3 et Cy5
- les différences de demi-vie et d'intensité du Cy3 et du Cy5
- les différences de quantités de sondes déposées par les différentes aiguilles, dans le cas où l'on utilise un robot spotteur à aiguilles
- une détection des signaux de fluorescence qui, sur une très large gamme, n'est pas proportionnelle aux quantités de molécules marquées
- les problèmes intrinsèques à l'analyse d'image

La normalisation est donc définie comme un processus de transformation des intensités, précédant toute analyse des données, et permettant l'étude de l'expression des gènes. Elle permet entre autre de comparer les ratios mesurés sur une même puce, ou de comparer les résultats obtenus avec plusieurs puces (étude temporelle par exemple).

Il existe de nombreuses techniques de normalisation, qui restent pour la plupart encore à évaluer (voir [Bilban et al 2002, Quackenbush 2002] pour un état de l'art). Park et al [2003] effectuent une comparaison de quelques unes de ces méthodes.

3.5.1 La normalisation par rapport à la moyenne globale des intensités

La méthode de normalisation la plus simple est celle qui consiste à supposer que les gènes étudiés sur la puce représentent un échantillon aléatoire de l'ensemble des gènes de l'organisme. Dans ce cas, la majorité des gènes s'expriment de la même façon dans les deux conditions d'expériences, et donc émettent un signal identique en rouge et en vert. Cette méthode consiste à ajuster les données afin que la moyenne des logarithmes des ratios soit égale à zéro sur l'ensemble de la puce. On calcule un coefficient multiplicateur N tel que, pour chaque gène i :

$$R_{r/v}(i)normalisé = N \times R_{r/v}(i)$$

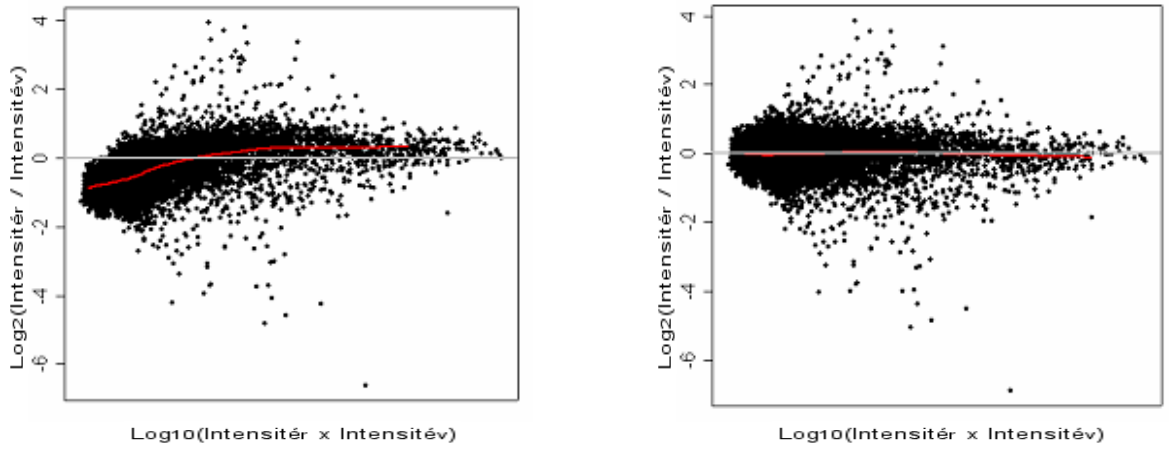
où $N = \frac{1}{2^{moyenne\ sur\ la\ puce(\log_2(R_{r/v}(i)))}}$

Il existe des variantes de cette méthode : on peut par exemple travailler sur la médiane des ratios plutôt que sur la moyenne. Il est également possible de calculer le facteur de normalisation en prenant en compte un certain nombre de gènes prédéfinis plutôt que l'ensemble des gènes. Ces gènes pourront par exemple être des gènes dont on sait à l'avance qu'ils s'expriment de manière constante durant l'expérience. Cette méthode est préférable lorsqu'il y a beaucoup de gènes exprimés de manière différentielle.

3.5.2 La normalisation « Lowess »

Plusieurs études ont montré qu'il existait un biais systématique dans les valeurs des logarithmes des ratios [Yang, Y.H. et al 2002, Yang, I.V. et al 2002]. Ce biais peut être observé en traçant le nuage de points ($x = \log_{10}(\text{Intensité}_r * \text{Intensité}_v)$, $y = \log_2(R_{r/v})$). Ce graphe, qui permet donc de visualiser les ratios en ordonnée en fonction du produit des intensités en abscisse, a tendance à s'incurver, notamment aux faibles intensités, plutôt que de rester centré sur la droite $\log_2(R_{r/v}) = 0$ (Figure 7-a). Ceci montre que les ratios présentent une dépendance systématique à l'intensité car le signal émis par un des fluorochromes décroît plus rapidement que l'autre.

La méthode de normalisation Lowess (Locally weighted scatter plot smoothing) a été proposée afin de corriger ce biais systématique. Cette méthode statistique permet, grâce à un système de « fenêtre glissante », de calculer une courbe de normalisation ajustée à la forme du nuage par régression linéaire locale (Figure 7-b).



(a) (b)

Figure 7 : Application de la méthode de normalisation Lowess

Le nuage de points ($x = \log_{10}(\text{Intensité}_r * \text{Intensité}_v)$, $y = \log_2(\text{Intensité}_r / \text{Intensité}_v)$) permet de visualiser le biais systématique des valeurs des ratios aux faibles intensités (a). Après application de la méthode de normalisation Lowess (b), le nuage est recentré sur la droite $\log_2(R_{r/v}) = 0$.

3.5.3 Autres méthodes de normalisation

La normalisation par intensité globale suppose que l'expérience est réalisée avec la même quantité d'ARN cible pour les deux échantillons, et que le nombre de molécules individuelles d'ARN dans les deux échantillons est approximativement le même. Ainsi, si la représentation

dans l'échantillon d'un ARN donné augmente, celle d'autres ARN doit diminuer, et le nombre total de molécules de chaque échantillon qui s'hybrident avec la puce est le même. On calcule un facteur de normalisation de façon à ce que les sommes des intensités de tous les spots soient égales pour les deux échantillons.

On peut également citer :

- La normalisation par régression linéaire
- Les méthodes par « invariant de rang » [Tseng et al 2001]
- Les techniques basées sur les statistiques des ratios [Kerr et al 2000, Wolfinger et al 2001]
- L'introduction dans le mélange cibles d'ARN exogènes (ou « RNA spikes ») de concentration connue

3.6 Analyse des données

La dernière étape d'une expérience de puce à ADN est certainement la plus difficile. Il s'agit d'exploiter les milliers de valeurs numériques produites afin d'en extraire une information biologique pertinente. Dans le cas d'une expérience classique de mesure du trancriptome, l'idée directrice est que si des gènes ont le même profil d'expression, ils ont des chances d'être impliqués dans une même réponse et éventuellement d'être régulés par le même mécanisme (« coupable par association »). La plupart des méthodes vont donc chercher à regrouper des gènes ayant le même profil d'expression. Les applications sont multiples, tant en recherche fondamentale (caractérisations des gènes de fonction inconnue, étude de la régulation de la transcription, détermination de réseaux géniques …) qu'en recherche appliquée (identification de cibles thérapeutiques, diagnostic/pronostic clinique, classification des pathologies…).

Il existe un très grand nombre de méthodes d'analyse de données. Parmi elles se trouvent les techniques classiques de classification supervisée ou non, ainsi que de nombreuses méthodes issues d'autres disciplines, transposées aux données de puces à ADN.

3.6.1 Classification

Chaque gène est considéré comme un individu, caractérisé par un ensemble de p paramètres (où p est le nombre d'expériences réalisées) et les valeurs de ces paramètres sont les niveaux d'expression du gène à l'expérience considérée [Slonim 2002]. L'objectif est alors de regrouper les individus les plus semblables au sein de nuages de points (les clusters), au moyen d'une métrique. La métrique utilisée, le type et les paramètres de regroupement définissent l'algorithme de classification [Jain and Dubes 1988]. Parmi les méthodes les plus utilisées en analyse de données d'expression, on trouve la classification hiérarchique [Eisen et al 1998, Spellman et al 1998], les nuées dynamiques (ou k-means), et les algorithmes de partition (minimisation d'une distance intra-classe et/ou maximisation d'une distance inter-classe) [Heyer et al 1999, De Smet et al 2002].

A titre d'exemple, voici les grandes lignes des deux algorithmes les plus utilisés.

Clustering hiérarchique (voir Figure 8):

1. Calculer une matrice de distances entre gènes.

2. Sélectionner les deux gènes les plus proches, et les remplacer dans la matrice par leur gène moyen. La liaison entre ces deux gènes fait une branche dans l'arbre, dont les deux gènes sont les feuilles.

3. Itérer le processus (1) jusqu'au dernier gène.

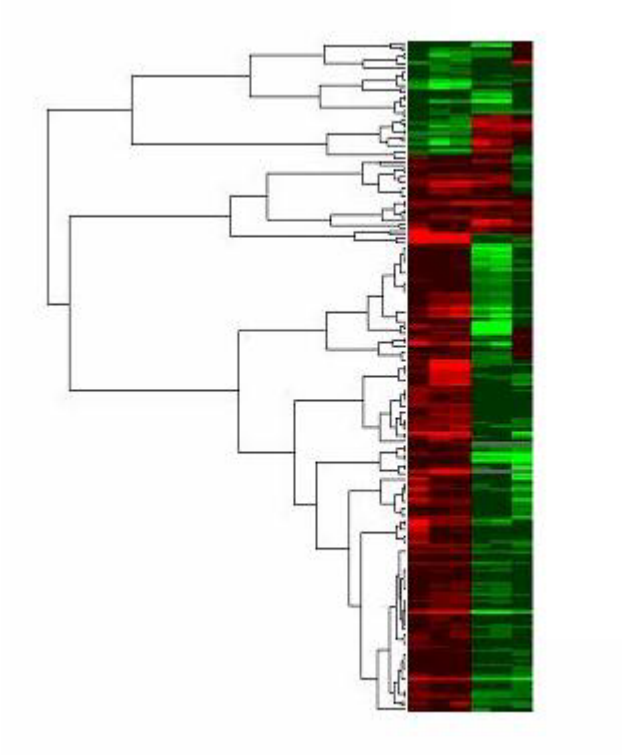


Figure 8 : Exemple de résultat de l'algorithme de clustering hiérarchique sur des données d'expression de gènes.

A droite, les données sont représentées par une matrice de rectangles colorés avec en colonnes les différentes expériences et en ligne les différents gènes étudiés). La valeur numérique du ratio d'expression est représentée sur une échelle de couleurs/niveaux de gris.
A gauche, l'arbre schématise les étapes de regroupement de l'algorithme de clustering hiérarchique.

Nuées dynamiques :

1. Choisir k gènes formant ainsi k clusters.

2. (Ré)affecter chaque gène G au cluster C_i de centre M_i tel que dist(G,M_i) est minimal.

3. Recalculer M_i de chaque cluster (le barycentre).

4. Aller à l'étape 2 si on vient de faire une affectation.

3.6.2 Analyse en composantes principales

Le but d'une Analyse en Composantes Principales est de résumer et de hiérarchiser l'information contenue dans la matrice de données d'expression constituée de n lignes (gènes) et de p colonnes (expériences). Il s'agit d'un changement de coordonnées dans l'espace permettant de représenter de façon plus synthétique les variations des données, sans les dénaturer. La méthode comporte plusieurs étapes :

1. standardiser les données : les données sont centrées et réduites

2. constituer la matrice de corrélation entre les variables. Cette matrice est carrée symétrique d'ordre p

3. trouver les vecteurs propres de la matrice de corrélation, ainsi que leur valeur propre associée.

4. calculer les coordonnées des gènes et des expériences sur ces vecteurs, pour la représentation graphique.

5. calculer les autres paramètres

Les vecteurs propres donnent les axes factoriels, et en choisissant deux axes, on peut représenter les gènes et les expériences sur un graphique en deux dimensions. Les deux premiers fournissent par convention le maximum d'information pour la représentation graphique. Cette méthode a été largement utilisée sur des données de puces à ADN [Raychaudhuri et al, 2000,Yeung and Ruzzo 2001].

3.6.3 Réseaux de Neurones

Un réseau de neurones est un réseau d'unités élémentaires (les nœuds) interconnectées, à fonctions d'activation linéaires ou non linéaires ([Hertz et al 1991, Haykin 1994], voir également [Anderson and Rosenfeld 1998] pour les articles historiques). Ces nœuds sont regroupés pour les réseaux multicouches en au moins deux sous-ensembles de neurones : un sous-ensemble d'entrée, un autre de sortie et éventuellement un ensemble de neurones cachés (voir Figure 9). De nombreux modèles de réseaux existent (réseaux de Hopfield, perceptrons multicouches,…), les différents nœuds étant complètement ou partiellement interconnectés aux autres. L'ensemble des liens convergeant vers un nœud constitue les connexions entrantes du nœud. Ceux qui divergent vers d'autres nœuds sont les connexions sortantes. A chaque connexion entre des nœuds i et j, est associé un poids w_{ij} représentant la force de l'influence du nœud i sur le nœud j. L'ensemble des poids est regroupé dans un vecteur de poids synaptiques w. Les poids des connexions sont éventuellement modifiés au cours d'une phase d'apprentissage. Modifier la sortie des nœuds à partir de leurs entrées consiste tout d'abord à calculer l'activation présente à l'entrée du nœud puis à calculer la sortie du nœud suivant la fonction d'activation qu'elle possède.

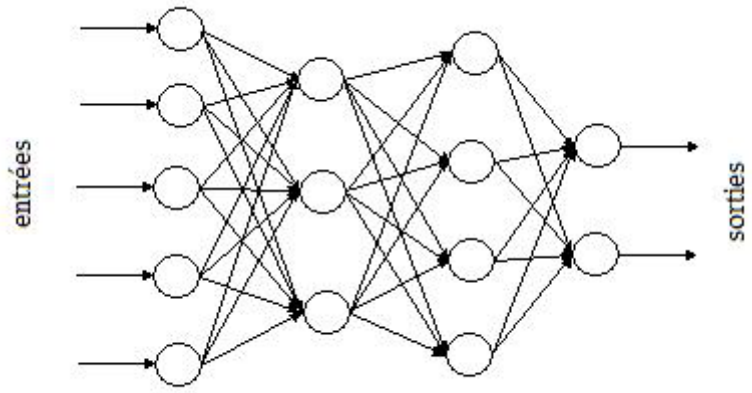


Figure 9 : Schéma d'un réseau de neurones de type perceptron multicouches.

Un réseau de neurones peut ainsi être défini pour chaque nœud par quatre éléments :

- La nature de ses entrées, qui peuvent être binaires ou réelles ;
- La fonction d'entrée totale *e*, qui définit le pré-traitement effectué sur les entrées.
- La fonction d'activation *f* du nœud qui définit son état de sortie en fonction de la valeur de *e*.
- La nature de ses sorties, qui peuvent être binaires ou réelles.

Deux éléments sont enfin nécessaires au bon fonctionnement du réseau : une fonction de coût et un algorithme d'apprentissage. L'apprentissage consiste en l'adaptation des paramètres du réseau de neurones pour donner une réponse désirée à une entrée donnée.

Un des réseaux de neurones les plus utilisés en analyse de données d'expression de gènes est le SOM (Self Organizing Maps), réseau non supervisé d'apprentissage [Kohonen 1997]. On parle également de cartes auto organisatrices de Kohonen. Le but du SOM est de trouver des vecteurs prototypes qui représentent les données d'entrée, tout en réalisant une bijection continue entre l'espace d'entrée et un maillage (ensemble de neurones de dimension donnée, facilement visualisable). Le principe de construction du SOM est itératif : après avoir choisi une géométrie du maillage (par ex. une grille 3*2), les nœuds sont mappés dans un espace à k dimensions, initialement aléatoirement, puis ajustés itérativement. A chaque itération, il y a sélection aléatoire d'un point et déplacement des nœuds dans la direction de ce point. Le nœud le plus proche est celui qui bouge le plus ; ainsi il y a agrégation de points en fonction de relations de voisinage. Cette méthode a été testée sur des données de levure [Tamayo et al, 1999, Toronen et al, 1999] et comparée aux méthodes hiérarchiques classiques [Herrero et al 2001].

3.6.4 Support Vector Machines (SVM)

Les machines à vecteurs de support (Support Vector Machine ou SVM) représentent une technique d'apprentissage supervisé, elles ont été utilisées avec succès dans de nombreux problèmes de reconnaissance de motifs comme la reconnaissance de texte, la reconnaissance de visage, l'identification vocale… [Burges, 1998]. Le principe général est une séparation d'un ensemble de données labellisées (ensemble de test) par un hyperplan maximisant la

distance aux points de test. Dans le cas où aucune séparation par un tel hyperplan n'est possible, il y a possibilité de coopération entre SVM et une technique de noyaux qui réalise une séparation non linéaire. Les données dont on ne connaît pas l'étiquetage sont ensuite testées par rapport à l'hyperplan séparateur, et donc classées (avec un intervalle de confiance dépendant par exemple de la distance à l'hyperplan). Dans le domaine de la biologie moléculaire, la phase d'apprentissage peut se faire avec les profils d'expressions de gènes dont les fonctions sont bien connues. Ensuite, étant donné un gène « inconnu », cette méthode permet de tester si ce gène appartient ou non à une classe donnée.

Les SVM commencent à être très utilisées pour classifier les données d'expériences de type microarrays. Dans [Brown et al 1997], les auteurs appliquent cette méthode aux données d'expression de la levure *Saccharomyces cerevisiae*. Ils comparent les SVM à plusieurs autres méthodes supervisées (Fenêtres de Parzen, discriminant linéaire de Fisher, arbres de décision), et testent différentes fonctions noyau. Dans [Pavlidis et al, 2002], les auteurs testent la capacité des SVM à apprendre sur des données hétérogènes (données d'expression + profil génétique de chaque gène). De nombreux travaux sont également menés dans le domaine de la classification de tissus cancéreux [Furey et al, 2000, Guyon et al, 2002].

3.6.5 Autres méthodes

De multiples méthodes issues d'autres disciplines scientifiques ont été testées sur les données de puces à ADN. Parmi elles, on peut citer les techniques basées sur la logique floue. Il s'agit d'interpréter les niveaux d'expression à l'aide d'une approche floue, qui permet de trouver par exemple des relations entre gènes. Globalement, chaque niveau d'expression ou groupe de niveaux lu est relié à un quantificateur flou (« grand », « petit », « moyen »), et un ensemble de règles floues est appliqué pour inférer un résultat (par exemple « si l'expression de G1 est grande et l'expression de G2 est faible, alors G3 est faiblement inhibé). L'utilisation de la logique floue dans l'interprétation des données issues des biopuces permet de prendre en compte les ambiguïtés des mesures (imprécisions dues au bruit lors de la mesure, imprécision lors de la détection des spots sur la plaque…). [Azuaje, 2001, Guthke et al, 2001, Woolf and Wang 2000].

De nombreuses méthodes statistiques et probabilistes sont également utilisées en analyse de données d'expression [Baldi and Long 2001, Kerr and Churchill 2001], ainsi que des techniques issues du domaine des bases de données [Lemkin et al 2000] et de la théorie de l'information [Fuhrman et al 00].

4 Les principales applications

4.1 Les études de transcriptome

La première utilisation des puces à ADN est historiquement l'analyse du transcriptome [DeRisi et al 1997], décrite dans les paragraphes précédents. Le transcriptome donne une représentation de l'état de la cellule dans des conditions données, et des processus biologiques qui s'y déroulent. Une telle étude permet d'obtenir le profil d'expression des gènes étudiés, c'est-à-dire la variation de leur niveau d'expression en fonctions de différents paramètres.

De très nombreuses études ont été réalisées sur différents organismes pour identifier les gènes co-régulés dans certaines conditions expérimentales spécifiques. Ainsi, chez la levure *Saccharomyces cerevisae*, des analyses de transcriptome ont été effectuées au cours du cycle cellulaire [Spellman et al 1998], au cours de la sporulation [Chu et al 1998], ou pour étudier l'influence du milieu extérieur [Kuhn et al 2001].

La technologie des puces à ADN trouve également de nombreuses applications dans le domaine médical. Elles sont notamment devenues un outil majeur dans la recherche sur le cancer, car elles permettent d'obtenir un profil de l'expression de l'ensemble des gènes d'une cellule à un instant donné, et de comparer le fonctionnement d'une cellule cancéreuse à celui d'une cellule saine [Segal et al 2005]. Les applications sont multiples :

- La classification de tumeurs : il s'agit d'un problème clé pour le développement de thérapeutiques nouvelles et individualisées [Golub et al 1999, Sorlie et al 2001].
- La réponse aux traitements médicamenteux des cellules cancéreuses [Kudoh et al 2000, Scherf et al 2000].
- Le diagnostic : l'étude des profils d'expression de cellules tumorales permet de rechercher des signatures d'expression associées par exemple au développement métastatiques ou à la résistance à certains traitements [van 't Veer et al 2002, Hedge et al 2001].

Enfin, les puces à ADN sont très utilisées en pharmacologie pour le développement de nouvelles molécules à visée thérapeutique [Crowther 2002]. Elles permettent entre autre de découvrir les fonctions des protéines codées par les gènes et donc d'identifier de nouvelles cibles potentielles pour les médicaments [Debouk 1999]. Elles permettent aussi de surveiller les modifications de l'expression des gènes en réponse à un traitement [Ivanov et al 2000]. Elles ouvrent la voie à des thérapies personnalisées en fonction du profil génétique du patient [Celis et al 2000]. La biopuce est aussi l'outil principal d'un domaine de recherche relativement récent : la toxicogénomique. C'est la génomique appliquée à l'identification des gènes affectés par l'exposition de la cellule à un produit chimique. La toxicogénomique étudie les relations entre la réponse d'une cellule à certaines substances chimiques (présentes dans l'environnement ou l'alimentation, certains médicaments…) et les changements dans l'expression des gènes de cette cellule [Chin and Kong 2002].

4.2 L'étude des réarrangements génomiques par CGH-array

La technique de CGH (Comparative Genomic Hybridization) permet de détecter les variations du nombre de copies de régions d'ADN génomique d'une cellule dans certaines conditions particulières [Kallioniemi et al 1992]. L'application la plus courante est l'étude des cancers, car il est fréquent d'observer des aberrations chromosomiques (augmentation du nombre de copies des oncogènes[2], diminution du nombre de copies des gènes suppresseurs de tumeur) lors de la progression d'une tumeur [Albertson, et al 2003].

La méthode de CGH-array est la combinaison de la CGH classique avec la technologie des puces à ADN. Elle permet d'effectuer des études avec une résolution beaucoup plus élevée, en déposant un grand nombre de spots qui cartographient des régions chromosomiques [Solinas-Toldo et al 1997, Pinkel et al 1998, Snijders et al 2001, Ishkanian et al 2004]. Dans les études sur le cancer, des cellules tumorales et des cellules de tissu sains sont marquées avec des fluorochromes différents et hybridées sur les lames CGH.

Cette technique est également très utilisée pour comparer deux génomes d'organismes proches afin d'évaluer leurs différences. Par exemple la comparaison d'une souche microbienne pathogène avec une souche non pathogène permet d'identifier les gènes de virulence. Cette approche a notamment été utilisée pour l'agent responsable de la tuberculose, la bactérie *Mycobacterium tuberculosis* [Behr et al 1999].

4.3 La détection des SNP

Les SNP (Single Nucleotide Polymorphism) constituent la forme la plus abondante de variations génétiques dans le génome humain. Il s'agit de mutations ponctuelles isolées (variation d'une seule base), se retrouvant le long de l'ensemble de l'ADN du génome. Il est possible de détecter les SNP avec des puces à ADN utilisant des oligonucléotides courts, suffisamment spécifiques pour discriminer des séquences différant d'un seul nucléotide.

Les variations de type SNP sont associées à de la diversité entre populations ou individus, mais également à une différence de sensibilité à certaines maladies et à la réponse individuelle aux médicaments. C'est pourquoi la détection des SNP par puces à ADN présente un intérêt en oncologie et en pharmacologie en vue du diagnostic et du suivi des traitements [Hirschhorn et al 2000].

4.4 La méthode du « ChIP-on-Chip »

A l'intérieur de la machinerie cellulaire, il existe de nombreuses interactions entre l'ADN et certaines protéines. Ces dernières se fixent à la molécule d'ADN, par exemple pour réguler l'expression des gènes (voir Annexe). La technique dite de « ChIP-on-Chip » (Chromatin-ImmunoPrecipitation on Chip) permet d'identifier les sites de fixation de protéines d'intérêt

[2] gène favorisant l'apparition et le développement d'une tumeur.

sur l'ADN génomique [Ren et al 2000, Lieb et al 2001, Buck and Lieb 2004]. Il s'agit d'une combinaison entre une méthode d'immunoprécipitation, qui permet d'isoler une protéine et son ADN associé, et la technique des puces à ADN.

Schématiquement, les interactions des protéines régulatrices à l'ADN sont stabilisées par un traitement au formaldéhyde. L'ADN génomique est alors fragmenté, et un anticorps spécifique de la protéine d'intérêt est utilisé pour reconnaître le complexe protéine-ADN. L'immunoprécipitation permet d'isoler cette protéine et les fragments d'ADN sur lesquels elle se fixe. Cet ADN purifié est amplifié, marqué par fluorescence, puis hybridé sur une puce sur laquelle sont spottées des séquences représentant une « carte » des chromosomes. On peut ainsi identifier les zones sur lesquelles va se fixer préférentiellement la protéine d'intérêt, et découvrir les gènes potentiellement régulés par cette protéine.

Cette méthode est surtout utilisée sur des organismes dont la taille des régions intergéniques est relativement faible, afin de faciliter la construction de puces cartographiant ces zones. En effet, les zones de fixation des protéines se situent principalement dans ces régions. De nombreuses expériences de « ChIP-on-Chip » ont été réalisées pour étudier les facteurs de transcription de la levure *Saccharomyces cerevisiae* [Iyer et al 2001].

4.5 La détection d'organismes

Une des applications des puces à ADN est également l'identification des organismes présents dans un milieu inconnu. Si l'on est capable de définir des séquences nucléiques spécifiques d'un organisme donné, appelées biomarqueurs, ces séquences peuvent constituer des sondes déposées sur une biopuce. Il est alors possible d'hybrider cette puce avec un mélange cible inconnu marqué par fluorescence, et de déterminer quels sont les organismes présents dans ce mélange. La technologie des puces à ADN permet d'effectuer de telles analyses très rapidement : une biopuce peut porter des dizaines de milliers de biomarqueurs et donc reconnaître autant d'organismes différents.

La principale application de cette technique est l'analyse de communautés microbiennes. En effet, les micro-organismes jouent un rôle fondamental dans les grands cycles biogéochimiques, l'agriculture, les biotechnologies ou la médecine, et on estime que la plupart des espèces microbiennes existantes n'ont pas encore été identifiées et décrites. L'identification de nouvelles espèces et la compréhension de leurs relations avec les écosystèmes sont des enjeux majeurs de la biologie actuelle.

Lors des dernières décennies, la plupart des études sur la diversité microbienne d'écosystèmes complexes ont été biaisées, non seulement par le fait que de nombreux micro-organismes sont non-cultivés, mais aussi à cause du défaut de sensibilité des méthodes d'identification microbiologiques traditionnelles [Amann et al 1995, Hugenhotlz et al 1998]. Dans la dernière décennie, l'utilisation de nouvelles techniques de biologie moléculaire a connu un essor considérable, présentant une alternative puissante d'identification des micro-organismes par rapport aux approches culturales. L'analyse de communautés microbiennes à l'aide des puces à ADN allie les avantages de ces techniques moléculaires classiques avec la possibilité d'identifier par une seule manipulation des milliers d'espèces. De nombreuses études ont été réalisées en utilisant l'ARN ribosomique comme biomarqueur (voir paragraphe 5.3.2) [Valinsky et al 2002, Wilson et al 2002, Zhang et al 2002].

5 Problématique biologique

5.1 Contexte des travaux et objectifs

La majeure partie de mon travail de thèse s'est déroulée en collaboration avec l'équipe Génomique Intégrée des Interactions Microbiennes dirigée par le professeur Peyret du Laboratoire de Biologie des Protistes (LBP, UMR CNRS 6023) de l'université Blaise Pascal. Cette équipe de biologistes souhaitant réaliser des expériences de puces à ADN (utilisant des sondes oligonucléotidiques), elle a développé une coopération avec le LIMOS afin d'approfondir les aspects informatiques de ces expériences et de ne pas se concentrer uniquement sur les aspects biologiques. En effet, même s'il est possible aujourd'hui, comme nous l'avons vu précédemment, de faire réaliser des expériences de puces à ADN par des sociétés spécialisées (qui prennent tout en charge depuis la fabrication des puces jusqu'à l'obtention du résultat), l'équipe du professeur Peyret souhaitait garder une maîtrise totale sur l'ensemble du processus. Dans le cadre de ces expériences, les biologistes du LBP utilisent donc des puces fabriquées « à façon » et réalisent eux-mêmes l'obtention des cibles, les hybridations et l'extraction des données brutes, en laissant seulement à la charge de sociétés extérieures la fabrication des lames. Du point de vue informatique, nous nous chargeons au laboratoire entre autres de l'analyse des images et de l'extraction d'information, en combinant l'utilisation de logiciels existants et le développement d'algorithmes originaux. Ainsi, de nouvelles méthodes ont été proposées dans le domaine de l'analyse d'images de puces à ADN [Barra 2006] et de l'analyse de données [Barra 2004].

Le travail de thèse que nous présentons ici porte sur une autre partie du processus : la conception de l'expérience. En effet, les premiers tests réalisés avec des puces comprenant un petit nombre de spots ont révélé l'importance de l'étape de conception des sondes (c'est-à-dire la détermination des séquences des oligonucléotides à déposer sur la lame). L'objet de cette thèse est donc le développement d'algorithmes pour la conception de sondes oligonucléotidiques dans le cadre d'expériences de puces à ADN.

Ces algorithmes doivent permettrent aux biologistes du LBP de concevoir des expériences pour deux problématiques distinctes :

1. l'étude globale de l'expression des gènes d'un parasite eucaryote intracellulaire obligatoire (*Encephalitozoon cuniculi*) au cours du cycle de développement.

2. la conception de biopuces dites phylogénétiques pour l'identification des micro-organismes présents dans un environnement donné.

Ces deux problématiques biologiques sont présentées dans les paragraphes suivants.

5.2 Etudes de l'expression des gènes d'un parasite eucaryote intracellulaire obligatoire

5.2.1 Les microsporidies

Le modèle d'étude des biologistes du LBP est un micro-organisme unicellulaire appartenant à l'ordre des microsporidies : *Encephalitozoon cuniculi*. Les microsporidies sont des parasites eucaryotes intracellulaires obligatoires, ils ont besoin d'envahir les cellules d'un organisme (l'hôte) pour se développer et se reproduire, causant ainsi des pathologies. Plus de 1200 espèces de microsporidies ont été identifiées et sont capables d'infester tout le règne animal [Vivarès and Méténier 2001]. Ces agents infectieux causent des pertes économiques importantes dans les élevages de vers à soie, d'abeilles, de poissons, de lapins et d'animaux domestiques. L'intérêt pour l'étude de ce pathogène s'est accru depuis la confirmation de son incidence en pathologie humaine suite à la pandémie du SIDA [Curry and Smith 1998]. De plus, des cas de microsporidioses sont apparus lors de traitements immunodépresseurs utilisés au cours de transplantations et de traitements anticancéreux [Weiss 2001], mais également chez des patients non immunodéprimés [Van Gool et al 2004].

Outre l'intérêt médical et vétérinaire, les microsporidies présentent des particularités les rendant attrayantes en terme de modèle biologique. Il s'agit d'eucaryotes unicellulaires dépourvus de Golgi typique, de mitochondrie, et donc dépendants énergétiquement de leurs hôtes. Le mécanisme d'infestation reste unique dans le monde vivant. Un tube polaire enroulé à l'intérieur de la spore infectieuse sera dévaginé suite à divers stimuli inconnus et permettra le passage du sporoplasme (noyau et cytoplasme) à l'intérieur de la cellule-hôte (Figure 10).

Le cycle de développement de ces pathogènes se déroule en deux phases. Tout d'abord la phase proliférative (mérogonie) a lieu soit au sein d'une vacuole parasitophore (genre *Encephalitozoon*), soit au contact direct du cytoplasme de l'hôte (genre *Enterocytozoon*). La deuxième phase appelée sporogonie permet la transformation des mérontes en sporontes avec l'apparition d'un revêtement dense aux électrons (future exospore des spores matures). La division des sporontes conduit à la formation des sporoblastes. Enfin, les sporoblastes se différencient en spores matures par acquisition de l'appareil invasif.

En terme d'évolution des génomes, ce groupe de parasites est particulièrement intéressant. En effet, il existe une plasticité importante de la taille des génomes avec des espèces du genre *Encephalitozoon* possédant des génomes inférieurs à 3 Mb alors que les génomes d'espèces du genre *Glugea* atteignent 19,5 Mb.

5.2.2 Un génome réduit et compact

Après une première phase de faisabilité de séquençage systématique du plus petit chromosome (218 kb) de l'espèce *Encephalitozoon cuniculi*, un projet collaboratif entre le LBP et le Centre National de Séquençage (Genoscope ; http://www.genoscope.fr) à Evry a été initié afin de séquencer la totalité du génome (2,9 Mb subdivisés en 11 chromosomes). Le projet a été mené à son terme (séquençage, assemblage, annotation) permettant d'obtenir des informations sur la structure originale des chromosomes microsporidiens (1 unité ADNr à chaque extrémité de tous les chromosomes avec une conservation sur près de 30 kb des extrémités télomériques et subtélomériques).

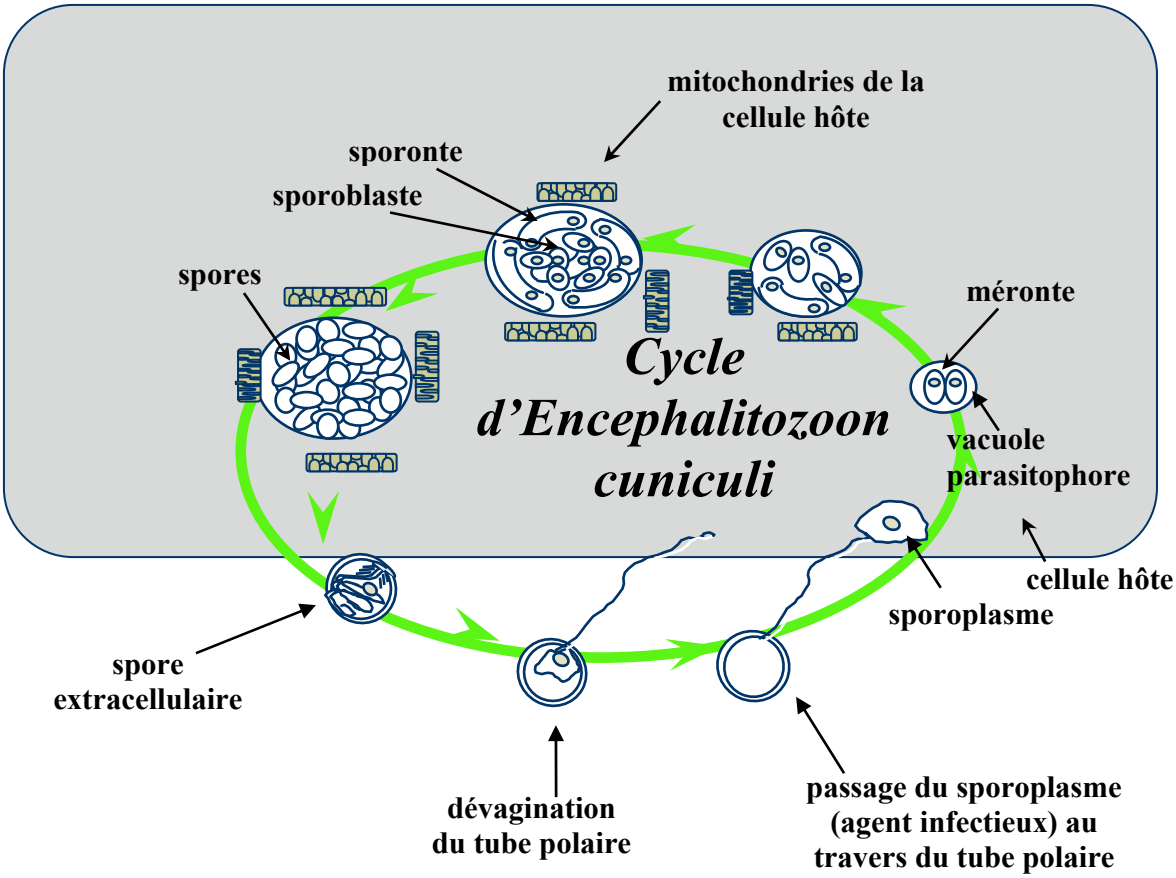


Figure 10 : Cycle de développement de la microsporidie *Encephalitozoon cuniculi*.

Les résultats de l'annotation du premier chromosome [Peyret et al 2001], puis du génome complet [Katinka et al 2001], font apparaître une forte compaction génique sur tous les chromosomes. En effet, il est fréquent de retrouver des régions intergéniques inférieures à 100 pb. Les gènes sont localisés au "cœur" des chromosomes, et les extrémités sub-télomériques, dont les séquences sont très conservées sur plus de 30 kb, contiennent une unité ADNr.

Au total 1997 CDS (« Coding DNA Sequence », séquence codante d'un gène) ont été identifiées et près de 40% de ces gènes codent pour des protéines à fonction connue. Les protéines impliquées dans la machinerie de réplication, de transcription, de traduction, de réparation et de dégradation sont fortement représentées. La machinerie d'épissage semble fonctionnelle mais seulement 13 introns ont été détectés dans le génome. Ce projet a permis de disposer pour la première fois de la totalité de l'information génomique d'un parasite eucaryote.

5.2.3 Conception de puces à ADN spécifiques

Afin de valoriser le travail de génomique réalisé sur *Encephalitozoon cuniculi* et pour continuer de décrypter la physiologie de ce pathogène pouvant servir de modèle pour d'autres parasites eucaryotes, le LBP a décidé d'utiliser la technologie des puces à ADN. Les résultats des expériences permettront également de mieux cerner les mécanismes de régulation de l'expression des gènes d'un eucaryote à génome minimal.

L'objectif est l'utilisation de puces ADN spécifiques du pathogène *Encephalitozoon cuniculi* pour identifier les différentes classes de gènes exprimés au cours du développement parasitaire. Pour ce faire, les biologistes réalisent une cinétique d'extraction des ARN messagers (cellules hôtes-parasites) après infestation des cellules hôtes par le parasite. Ainsi grâce à une cinétique précise et l'utilisation des puces à ADN nous serons capables de classer les gènes en fonction du cycle de développement. Cette approche globale très informative permettra également d'orienter, dans un certain nombre de cas, vers une fonction putative des protéines à fonction encore inconnue (co-expression avec des protéines à fonction connue; annotation relationnelle). De plus, l'analyse de ces résultats permettra de mettre en lumière des régulations particulières de l'expression des gènes microsporidiens. Ainsi, des organisations polycistroniques, des régulations antisens (dues à la compaction extrême du génome) et des facteurs de transcription spécifiques de différentes classes de gènes seront plus facilement identifiés. Une meilleure connaissance de la biologie d'un parasite intracellulaire sera alors possible et des cibles thérapeutiques pourront également être identifiées.

L'équipe Génomique Intégrée des Interactions Microbiennes souhaitait garder une maîtrise totale sur l'ensemble de ces expériences, elle souhaitait notamment déterminer elle-même les séquences des sondes à utiliser pour cibler les gènes du parasite. De plus, la spécificité d'une telle étude est que lors de la préparation des cibles, il est impossible de séparer les transcrits du parasite des transcrits de la cellule hôte, en l'occurrence des transcrits humains dans notre cas. Au cours de ma thèse, j'ai donc travaillé sur les aspects informatiques de la conception d'oligonucléotides spécifiques pour les expériences de puces à ADN, et sur le développement d'algorithmes adaptés à de telles études hôte-parasite.

5.3 Conception de biopuces phylogénétiques pour suivre l'évolution de communautés bactériennes lors d'un procédé de bioremédiation

5.3.1 La bioremediation

L'accumulation de produits chimiques toxiques dans l'environnement, depuis plusieurs décennies, est de plus en plus importante. Les nuisances et les risques pérennes pour les individus ainsi que pour les écosystèmes sont alors préoccupants [Lars 2002]. Les sols, en raison de leur position d'interface dans l'environnement, jouent un rôle essentiel dans les cycles biogéochimiques et dans le devenir des substances polluantes. Ils constituent un écosystème à part entière, réservoir d'une biodiversité considérable et encore faiblement connue [Amann et al 1995, Pepper et al 2002]. Ils sont également les supports de la production végétale, donc déterminants essentiels de la sécurité alimentaire, et influent directement sur la qualité de l'eau. Ainsi, les sols constituent un patrimoine dont la gestion durable doit s'imposer dans les années à venir comme une préoccupation de plus en plus forte. Afin de préserver ce patrimoine et de mieux connaître le fonctionnement de cet écosystème, il est nécessaire de développer des outils performants de contrôle et de suivi de la qualité des sols, ainsi que des méthodes fiables de réhabilitation pour les terrains contaminés.

Jusqu'alors, les tentatives de dépollution passaient surtout par des techniques physiques et/ou chimiques très invasives pour les sites traités [Rodgers and Bunce 2001]. En effet, ces méthodes pouvaient nécessiter l'excavation des terres à traiter et/ou entraîner des rejets de

produits potentiellement toxiques. Actuellement des solutions alternatives de décontamination voient le jour. Elles utilisent les capacités naturelles de certains organismes à dépolluer les sols, comme les plantes pour la phytoremédiation et les micro-organismes pour la bioremédiation. La bioremédiation est un ensemble de techniques consistant à augmenter la biodégradation ou la biotransformation, en inoculant des micro-organismes spécifiques (bioaugmentation) ou en stimulant l'activité de populations microbiennes indigènes (biostimulation) par apport de nutriments et par ajustement des conditions de milieu (potentiel d'oxydoréduction, humidité).

L'optimisation de la bioremédiation nécessite une connaissance plus approfondie des micro-organismes et en particulier des communautés bactériennes des sols, qui sont denses et diversifiées [Widada et al 2002]. C'est pourquoi l'équipe Génomique Intégrée des Interactions Microbiennes développe des techniques permettant d'améliorer les connaissances sur ces communautés microbiennes et de suivre leur évolution lors d'un procédé de bioremédiation.

5.3.2 Analyse des communautés bactériennes du sol à l'aide du biomarqueur ARNr 16S

Les ARN ribosomiques (ARNr) jouent un rôle fondamental dans la synthèse protéique. En effet, en association avec des protéines ribosomales, ces ARN ribosomiques formeront la machinerie de traduction, les ribosomes (voir Annexe). Les ARNr sont des molécules universelles, et dont les fonctions ont été établies très précocement dans l'évolution et n'ont pas été modifiées par les changements environnementaux des organismes. Les séquences nucléotidiques des ADNr présentent donc des régions hautement conservées et communes à toutes les bactéries, tandis que d'autres régions sont spécifiques d'une espèce. Ces molécules sont utilisées comme outils pour les études phylogénétiques des procaryotes, car les divergences entre des séquences d'ADNr pour deux organismes différents sont représentatives des variations entre les deux.

Les trois molécules d'ARNr sont classées en fonction de leur coefficient de sédimentation pendant une ultracentrifugation (23S, 16S et 5.8S). Leur longueur est respectivement de 3000, 1500 et 120 bases. C'est le gène codant pour l'ARN 16S qui est l'outil principalement utilisé pour l'identification moléculaire des bactéries ainsi que pour les études phylogéniques car celui-ci présente de nombreux avantages [Olsen et al 1986]. Il possède des régions hautement conservées, facilitant leur isolement, ainsi que des régions variables qui permettent de différencier les espèces.

Ces gènes sont souvent présents en plusieurs copies par cellules, étant naturellement amplifiés, ce qui rend la détection plus facile et la sensibilité de détection plus importante [Hugenholtz et Pace 1996, Guschin et al 1997]. Comme ces gènes ne subissent pas de transferts horizontaux et que l'évolution de leur séquence se fait de façon relativement lente, par rapport aux autres gènes procaryotiques, ils sont de bons marqueurs phylogénétiques.

L'ADN est le plus souvent étudié afin d'analyser les communautés bactériennes. Cette méthode peut-être à l'origine d'un biais car elle n'est pas représentative de la viabilité et de l'activité métabolique des micro-organismes. Les analyses basées sur l'ARN sont plus appropriées pour étudier les communautés métaboliquement actives [Felske et al 1997] mais paradoxalement moins utilisées du fait des difficultés causées par la difficulté d'obtention de matériel biologique de qualité.

5.3.3 Mise au point d'une biopuce à ADN oligonucléotidique

Les approches de biologie moléculaire précédemment développées permettaient seulement une évaluation limitée de la diversité microbienne spatio-temporelle de communautés complexes. Ces technologies sont :

- FISH : Fluorescent In Situ Hybridisation
- in-situ PCR : in-situ Polymerase Chain Reaction
- T-RFLP : Terminal-Restriction Fragment Length Polymorphism
- séquençage d'ITS : Internal Transcribed Spacer
- ARDRA : Amplified rDNA Restriction Analysis
- SSCP : Singled-Strand Conformation Polymorphism
- DGGE : Double Gradient Gel Electrophoresis
- TGGE : Temperature-Gradient Gel Electrophoresis

La technologie biopuce à ADN permet de suivre simultanément l'évolution de plusieurs centaines voire de plusieurs milliers de micro-organismes différents.

L'objectif de l'équipe Génomique Intégrée des Interactions Microbiennes est donc de mettre au point une biopuce ADN oligonucléotidique, afin de suivre l'évolution des principales communautés bactériennes lors d'un procédé de bioremédiation. Actuellement, la plupart des micro-organismes du sol étant encore non identifiés, il paraît peu judicieux d'utiliser uniquement des sondes complémentaires spécifiques d'espèces connues, ce qui rendrait la biopuce trop spécifique sans possibilité réelle d'identifier de nouvelles espèces actives. La stratégie est donc de concevoir à terme, des sondes capables de cibler des groupes de micro-organismes au niveau de l'ordre, de la famille et du genre. La biopuce permettra de connaître les groupes qui varient lors de la bioremédiation et donc ceux qui ont potentiellement un rôle dans ces processus (populations stables ou en augmentation). Des techniques de biologie moléculaire classiques seront ensuite mises en oeuvre afin d'identifier tous les micro-organismes de ces groupes : extraction des acides nucléiques du sol, amplification des gènes codant les ARNr 16S grâce à des amorces spécifiques de groupe, clonage des produits PCR obtenus et leur séquençage systématique.

Nous présentons dans ce manuscrit le développement d'algorithmes de conceptions de sondes oligonucléotidiques pour des biopuces destinées à suivre l'évolution de communautés microbiennes. Ce problème, tout en présentant des similitudes avec celui de la conception de sondes pour les biopuces de type transcriptome, montre également des particularités. En effet, il s'agit de déterminer des séquences spécifiques d'un organisme (ou d'un groupe d'organismes) parmi l'ensemble des séquences d'ARNr 16S connues.

6 Conclusion

Dans ce chapitre, nous avons présenté le cadre général de cette thèse, à savoir les expériences de puces à ADN. Nous avons vu que la réalisation de telles expériences nécessitait la mise en œuvre d'une succession d'étapes, tantôt biologiques, tantôt informatiques, relativement complexes. L'ensemble des étapes informatiques, que l'on pourrait regrouper sous le terme « bioinformatique des puces à ADN » fait appel à des disciplines très différentes telles que l'algorithmique, le traitement d'image, les bases de données, la statistique, ou la fouille de données.

Nous avons ensuite exposé le contexte biologique de nos travaux : d'une part l'étude globale de l'expression des gènes d'un parasite eucaryote intracellulaire obligatoire, et d'autre part la conception de biopuces dites phylogénétiques pour l'identification des micro-organismes présents dans un environnement donné. Dans les deux cas, la conception des biopuces nécessite une attention toute particulière dans le choix des sondes à déposer sur le support solide. Nous avons donc défini la problématique centrale de cette thèse : le développement d'algorithmes pour la conception de sondes oligonucléotidiques dans le cadre d'expériences de puces à ADN.

Dans le chapitre suivant, nous décrivons précisément ce qu'est le problème de conception de sondes et nous réalisons un inventaire des différentes méthodes de résolution existantes. Nous présentons également d'un point de vue pratique l'utilisation des logiciels implémentant ces méthodes.

Chapitre II

Etat de l'art : La conception d'oligonucléotides pour puces à ADN

1 Le problème de la détermination des oligonucléotides

1.1 Introduction

Le problème de la détermination des séquences à utiliser pour les sondes est un point crucial dans la chaîne de traitements que constitue l'expérience de puces à ADN. En effet, tous les efforts fournis pour optimiser les différentes phases d'utilisation de la puce (choix du support, conditions d'hybridation, préparation des cibles, lecture et analyse des données) seront inutiles si, à la base, les sondes nucléiques n'ont pas été choisies avec pertinence.

Quelque soit l'application envisagée, le but recherché reste le même : identifier une sonde spécifique s'hybridant de manière optimale avec le transcrit correspondant. Le problème revient à chercher une sonde optimale présentant la plus faible énergie libre d'hybridation pour le transcrit visé et le maximum d'énergie libre d'hybridation pour tous les autres transcrits. Cette énergie d'hybridation (ΔG) dépend de nombreux paramètres tels que les caractéristiques physico-chimiques du support, la structure secondaire et la concentration des ADNc, aussi il apparaît impossible de calculer exactement cette énergie, à l'heure actuelle. Cependant cette énergie peut être approchée de façon empirique en sélectionnant les sondes potentielles sur des critères thermodynamiques et expérimentaux.

Toutes les approches existantes sont donc basées sur le même principe : étant donnée une séquence nucléique S, trouver une sous séquence q de S qui satisfait au mieux un certain nombre de critères. Nous reviendrons par la suite sur la notion de « satisfaction au mieux d'un critère ». Cette recherche de séquence d'oligonucléotide est à répéter pour chaque gène étudié sur la puce.

Dans l'utilisation classique des puces à ADN (étude de l'expression des gènes), les séquences S utilisées sont les Coding Dna Sequences (CDS) des gènes. En effet, ces dernières sont complémentaires des séquences ADNc présentes dans le mélange cible, et peuvent donc être choisies comme sondes.

Les principaux critères que doit satisfaire l'oligonucléotide concernent :

- la spécificité de la séquence,
- la température de fusion,
- la composition en bases de la séquence,
- la structure secondaire,
- la position de l'appariement de l'oligonucléotide dans la séquence CDS.

1.2 Spécificité de la séquence

Lors de la phase d'hybridation, chaque sonde oligonucléotidique est mise en contact avec le mélange cible qui contient un très grand nombre de séquences. Dans le cas d'une étude d'expression des gènes, le mélange cible contient l'ensemble des transcrits des cellules étudiées dans les conditions de l'expérience. Il est donc nécessaire qu'une sonde donnée s'hybride uniquement avec sa séquence cible, et pas avec d'autres séquences présentes dans le mélange. On dira alors que la séquence de l'oligonucléotide est spécifique de sa cible. On parlera d'« hybridation croisée » lorsqu'une séquence non-cible vient s'hybrider à une sonde oligonucléotidique (Figure 11).

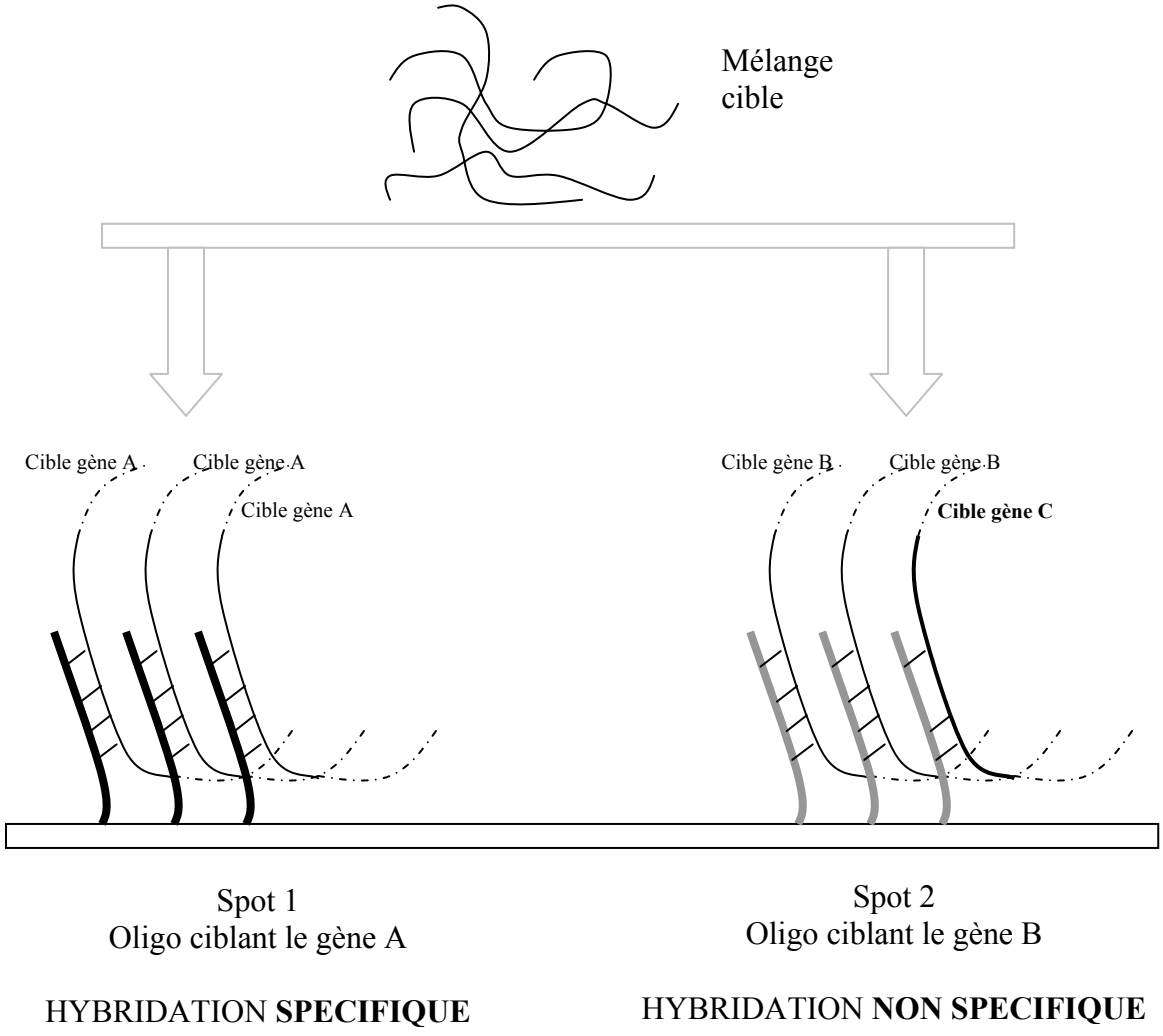


Figure 11 : Différence entre hybridation spécifique et non-spécifique.

Une séquence non-cible pourra s'hybrider avec une sonde si les deux séquences sont parfaitement identiques, mais également si elles présentent un petit nombre de substitutions de bases. Il faut donc connaître les conditions dans lesquelles une séquence donnée du mélange cible est susceptible de s'hybrider avec une sonde. Très peu d'études expérimentales ont été effectuées à ce sujet, et une seule fait référence dans la littérature. Il s'agit d'une étude sur la

spécificité des oligonucléotides de 50mers [Kane et al 2000]. Elle définit deux conditions pour qu'une sonde soit spécifique :

1) La séquence de l'oligonucléotide ne doit pas présenter plus de 75% de similarité (sur toute la longueur de la séquence) avec une séquence non-cible présente dans le mélange d'hybridation.

2) La séquence de l'oligonucléotide ne doit pas contenir une sous séquence de plus de 15 bases consécutives strictement identique à une séquence non-cible présente dans le mélange d'hybridation.

Dans la suite, nous désignerons ces deux conditions par « critère de Kane ». La plupart des logiciels de conception d'oligonucléotides existants s'appuient sur ce critère pour évaluer la spécificité.

Cette nécessité d'éviter les hybridations croisées est certainement le point le plus important dans la détermination des sondes, et donne toute sa difficulté au problème. Pour la levure *Saccharomyces cerevisiae*, une étude a montré que 253 CDS (4.5% de l'ensemble des CDSs) ne pouvaient pas être représentées par une unique sonde oligonucléotidique [Talla et al 2003]. De plus, ce pourcentage augmente avec la complexité du modèle biologique considéré.

1.3 Température de fusion

La température de fusion (Melting Temperature en anglais ou Tm) est la température à laquelle la moitié de l'ADN est sous forme monobrin et l'autre moitié sous forme double brin. Deux molécules simple brin s'hybrident en une molécule double brin, une molécule double brin se dénature en deux molécules simple brin. Le passage d'une forme à l'autre est brutal du fait du caractère coopératif de la réaction, que ce soit dans un sens ou dans l'autre. Ce passage se visualise très bien si on mesure la concentration de l'ADN par spectrophotométrie, car l'ADN simple brin absorbe plus les UV que l'ADN double brin. On mesure la Densité Optique (DO) à 260 nm, les bases puriques et pyrimidiques absorbant fortement dans l'ultraviolet à 260 nm (Figure 12).

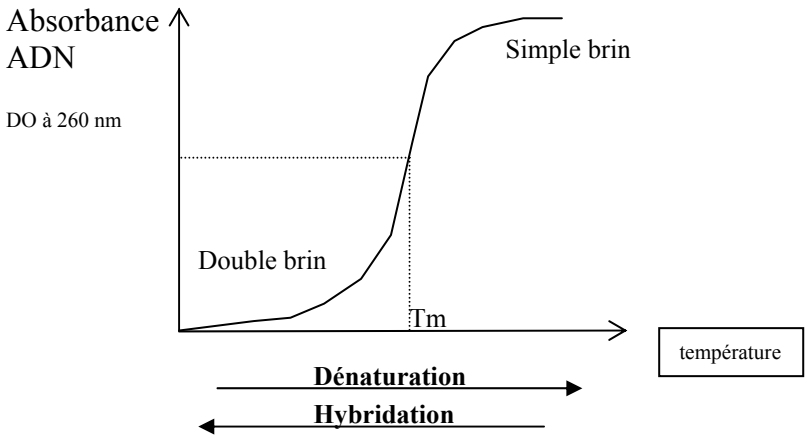


Figure 12 : Mise en évidence de la température de fusion par évaluation de l'absorbance en UV de l'ADN.

La température de fusion dépend de nombreux facteurs tels que la longueur du fragment d'ADN considéré, sa richesse en cytosines et guanines et la concentration en ion Na^+ du milieu réactionnel. En pratique, l'expérimentateur peut créer ou supprimer l'hybridation moléculaire en choisissant une température du milieu réactionnel inférieure, égale ou supérieure à la Tm.

La formule la plus simple pour estimer la Tm prend en compte uniquement la composition en bases de la séquence [Wallace et al 1979] :

$$T_m = 2(\#A + \#T) + 4(\#G + \#C) \qquad (°C)$$ [3]

Cette formule est très approximative, elle a été développée pour des oligos cours (< 25mers).

On peut également estimer la Tm par la méthode dite du « pourcentage de GC » (%GC) [Meinkoth and Wahl 1984] :

$$T_m = 81.5 + 16.6\log[Na^+] + 41(X_G + X_C) - \frac{500}{L} - 0.62F \quad (°C)$$

Où [Na+] est la concentration en sels, X_G et X_C les fractions molaires de G et C dans l'oligo, L la longueur de la séquence et F le pourcentage de formamide dans la solution.

[3] *#N* désigne le nombre d'occurrences de la base *N* dans la séquence

Là encore, cette formule donne une estimation de la température de fusion, mais la méthode la plus employée est celle des plus proches voisins. Cette dernière est considérée comme la méthode la plus fiable, elle ne prend pas seulement en compte la composition en base, mais les paramètres thermodynamiques attribués à chaque paire de bases dépendent des bases voisines [Wetmur 1991]. Ainsi, des oligonucléotides ayant la même longueur et la même composition en bases, mais des séquences différentes, auront des T_m différentes. La formule utilisée est alors :

$$T_m = \frac{\Delta H}{\Delta S + R\ln(C_T / 4)} + F([Na^+]) - 273.15 \qquad (°C)$$

Où : ΔH : enthalpie de formation de l'hélice d'ADN
ΔS : entropie de formation de l'hélice d'ADN
R : constante des gaz parfaits (1.987 cal/°c/mol)
C_T : concentration de l'oligonucléotide
F : terme de correction en fonction de la concentration en sels

Les paramètres utilisés pour calculer ΔH et ΔS sont donnés dans [Santalucia 1998].

Dans le cas d'une expérience de puce à ADN, l'ensemble des réactions d'hybridation entre chaque sonde et sa cible se déroule en même temps, dans le même milieu réactionnel, et donc dans les mêmes conditions. Ainsi, tous les couples sonde-cible doivent avoir une température de fusion voisine, pour assurer que toutes les réactions d'hybridation se déroulent correctement.

1.4 Composition en bases de la séquence – Notion de complexité

Pour définir si un oligonucléotide peut constituer une sonde pour un gène donné, il est également nécessaire d'étudier sa composition en bases.

Tout d'abord, on privilégiera une forte concentration en GC. En effet, lors de la formation d'une molécule d'ADN double brin, la paire G-C implique 3 liaisons hydrogène contre 2 pour A-T. Un appariement G-C a donc une énergie de dissociation plus grande : 5.5 kcal/paire contre 3.5 kcal/paire pour un appariement A-T. Ainsi, Plus une séquence d'ADN a un pourcentage de paires G-C élevé, plus il faudra fournir d'énergie pour la dénaturer.

Ensuite, il peut être intéressant d'interdire la présence de certains motifs dans la séquence de la sonde. Par exemple, une sonde comportant une suite de bases consécutives identiques (TTTTT…, AAAAA…, …) aura plus de chance de présenter des hybridations non spécifiques.

D'une manière plus générale, on cherchera à éviter les séquences de faible complexité. La notion de complexité d'une séquence nucléique, élément clé dans l'analyse des génomes aujourd'hui, a d'abord été définie de manière intuitive : les régions de faible complexité sont des séquences présentant des répétitions de motifs simples [Hancock 2002]. Il est important de noter que ces régions présentent un intérêt très important dans l'analyse des génomes, car elles peuvent correspondre à des fonctions communes. A l'inverse, pour la problématique de la conception d'oligonucléotides pour puces à ADN, on cherchera à éviter la présence de

telles régions dans les sondes, car sinon ces dernières risquent de présenter une faible spécificité.

Il existe de très nombreuses approches permettant d'évaluer la complexité d'une séquence d'ADN, issues de la théorie de l'information ou de la linguistique [Lempel and Zif 1976, Wootton 1996, Trifonov 1990, Troyanskaya et al 2002]. La séquence est considérée comme une suite finie de symboles pris dans l'alphabet {A, T, C, G}. L'une des plus utilisée est appelée complexité de Lempel et Zif [Lempel and Zif 1976], elle définit la complexité comme le nombre minimum d'étapes nécessaires pour générer la séquence, étant donné un certain nombre d'opérations élémentaires (génération d'un nouveau symbole, copie d'un fragment existant). Cette approche est une implémentation de la définition générale de la complexité d'un texte donnée par Kolmogorov[4] [Kolmogorov 1965], et est très utilisée dans le domaine de l'analyse des génomes [Gusev et al 1999].

Parmi les autres méthodes, certaines sont basées uniquement sur la fréquence des nucléotides, comme celle utilisée par le programme BLAST (notée *CWF*) lors de sa phase de prétraitement pour masquer les régions de faible complexité [Wootton 1996], ou comme l'entropie de Shannon (notée *CE*):

$$CWF = \frac{1}{N}\log_K\left(\frac{N!}{\prod_{i=1}^{K} n_i!}\right)$$

$$CE = -\sum_{i=1}^{K}\frac{n_i}{N}\log_K\left(\frac{n_i}{N}\right)$$

où N est la taille de la séquence, K la taille de l'alphabet (4 pour l'ADN) et n_i le nombre de symboles i dans la séquence.

Enfin les approches issues de la linguistique définissent la complexité d'une séquence par rapport à la richesse de son vocabulaire [Trifonov 1990, Troyanskaya et al 2002]: combien de mots différents de longueur i apparaissent-ils dans la séquence ? Ceci conduit aux deux variantes suivantes (notées *CT* et *CL*) :

$$CT = \prod_{i=1}^{N}\frac{V_i}{V_{\max,i}}$$

$$CL = \frac{\sum_{i=1}^{N} V_i}{\sum_{i=1}^{N} V_{\max,i}}$$

où N est la taille de la séquence, V_i est le nombre de mots différents de longueur i, et $V_{max,i}$ le nombre maximum de mots de longueur i que l'on peut trouver dans une séquence de longueur N.

[4] La complexité de Kolmogorov d'une chaîne de caractères est la taille du plus petit programme capable de l'engendrer.

1.5 Structure secondaire d'un oligonucléotide

Tout comme un ADN (ou ARN) simple brin possède la capacité de s'hybrider avec un brin de séquence complémentaire, il peut également se replier sur lui-même et s'hybrider avec ses propres bases. On nomme structure secondaire d'une séquence nucléique la description des appariements de bases à l'intérieur d'un même brin [Zucker 2000]. En terme de quantité d'information sur une séquence, cette représentation se situe entre la structure primaire, qui est uniquement la connaissance de la suite de bases et la représentation en trois dimensions de la molécule (Figure 13).

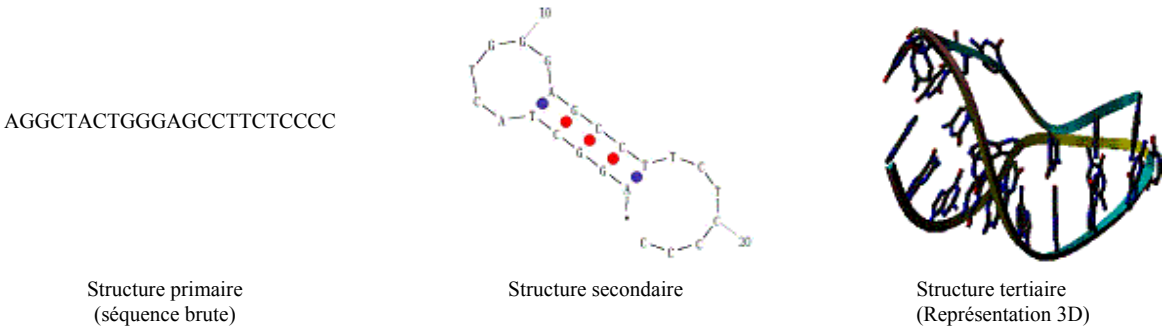


Figure 13 : Différentes représentation d'une séquence nucléique.

(Dans une séquence de longueur *n*, les bases sont numérotées de 1 à *n* à partir de l'extrémité 5')

Une sonde fixée sur une puce à ADN qui posséderait une structure secondaire stable dans les conditions d'hybridation perdrait la capacité de s'hybrider avec sa cible. C'est pourquoi on peut chercher à déterminer les structures secondaires possibles d'un oligonucléotide avant de le sélectionner comme sonde d'un gène donné.

Une séquence d'ADN est notée :

$D = d_1, d_2, \ldots, d_n$
où d_i est le ième nucléotide de la séquence (d_i appartient à l'ensemble {A, C, G, T}).
i se réfèrera donc à la ième base de la séquence.

Une **structure secondaire** ou « folding » sur D est un ensemble S de paires de bases (d_i,d_j), notées de manière simplifiée (i, j) , avec $1 \leq i < j \leq N$

L'ensemble S doit satisfaire (voir Figure 14) :

1. Si (i, j) appartient à S alors j - i > 3
(distance minimale entre deux bases pour qu'elles puissent s'apparier)

2. Si (i, j) et (i', j') sont 2 paires de bases, (en supposant $i \leq i'$ sans introduire de restriction), alors :

(a) i = i' et j = j' (c'est la même paire de bases), ou

(b) i < j < i' < j' ((i, j) précède (i', j')), ou

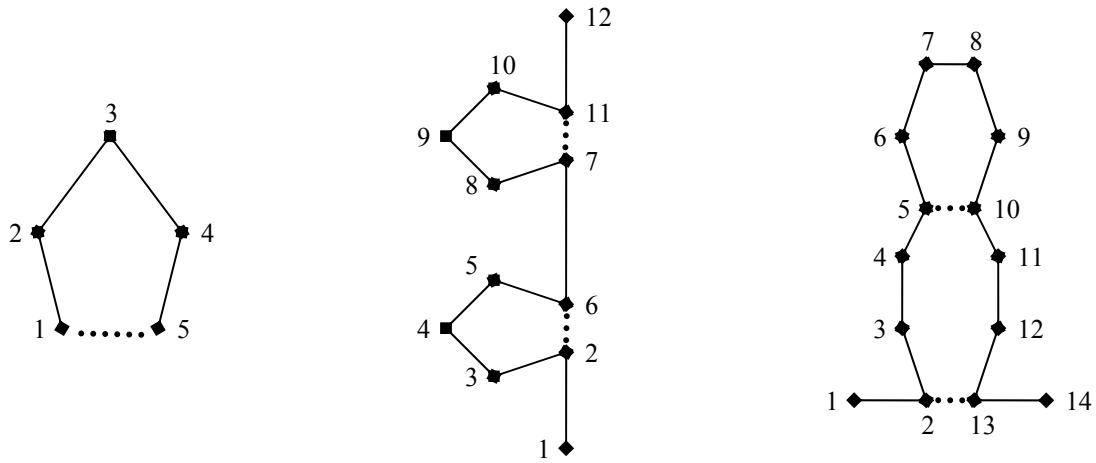
(c) i < i' < j' < j ((i, j) inclut (i', j')).

Remarque : La condition 2 exclut ce qu'on appelle les « pseudonoeuds », lorsqu'il existe deux paires de bases (i, j) et (i', j') telles que i < i' < j < j'. Ces configurations sont exclues car les méthodes de prédiction de structure secondaire par minimisation d'énergie ne peuvent pas les traiter. De plus, les pseudonoeuds sont souvent considérés comme des structures tertiaires.

Par exemple, la structure secondaire de la séquence de la Figure 13 est notée :

D = {(1,16) (2,15) (3,14) (4,13) (5,12)}

Déterminer la structure secondaire d'une séquence nucléique, c'est calculer tous les ensembles D possibles dans des conditions expérimentales données.



(1) (2) (b) (2) (c)

Figure 14 : illustration des différents critères que doit satisfaire une structure secondaire.

Critère (1) : dans cet exemple, S = { (1,5) }. On a bien 5-1>3.
Critère (2) (b) : S = { (2,6) (7,11) }. On a 2<6<7<11
Critère (2) (c) : S = { (2,13) (5,10) }. On a 2<5<10<13

1.6 Autres critères

D'autres critères peuvent également être utilisés pour la sélection des oligonucléotides, mais évidemment plus on prend en compte de critères, plus il sera difficile de trouver une séquence les satisfaisant tous.

Dans le cas où les cibles sont obtenues par une transcription inverse des ARNm en présence d'amorces oligo(dT)[5], il peut être intéressant de choisir une séquence la plus proche possible de l'extrémité 3' du CDS. En effet, la transcription inverse peut être interrompue avant la fin et dans ce cas là, les séquences proches de l'extrémité 5' seront absentes, ou présentes en moindre quantité dans le mélange cible.

Concernant le choix de la longueur des oligonucléotides, celui-ci est généralement effectué par le biologiste au tout début de la conception de l'expérience, et c'est ce choix qui détermine ensuite la sélection sur les autres critères. Les oligonucléotides longs (>50 bases) permettent de trouver un bon compromis entre spécificité et sensibilité.

2 Algorithmes pour la conception d'oligonucléotide pour puces à ADN

2.1 Evaluation des différents critères

Etant donné un CDS de longueur n, le problème de la détermination d'un oligonucléotide de longueur k consiste à déterminer une sous séquence du CDS de longueur k qui satisfasse un certain nombre de critères. Il existe deux manières de traiter ces critères : soit on considère une réponse de type oui/non, soit on définit des fonctions de score pour chacun d'entre eux.

Dans le cas d'une réponse de type oui/non, on définit une condition à satisfaire pour chaque critère considéré. Alors une séquence pourra constituer une sonde si et seulement si elle satisfait toutes les conditions définies [Rouillard et al 2002].

Exemple : l'ensemble des conditions à satisfaire pour un oligonucléotide o peut-être :

- spécificité : o doit satisfaire le critère de Kane par rapport à l'ensemble des séquences du mélange cible
- température de fusion : $70°C < Tm(o) < 80°C$

[5] Cette technique de rétrotranscription de l'ARN utilise une amorce constituée d'une suite de bases T s'hybridant avec la queue polyA des ARN. L'ADNc obtenu est donc synthétisé à partir de l'extrémité 3' de l'ARNm.

- composition en bases : %GC(o) > 60%
  CL(o) > 0.2
- structure secondaire : pas de structure secondaire stable de Tm > 60°C

Cette approche conduit à l'algorithme de base pour la recherche d'oligonucléotides présenté Figure 15.

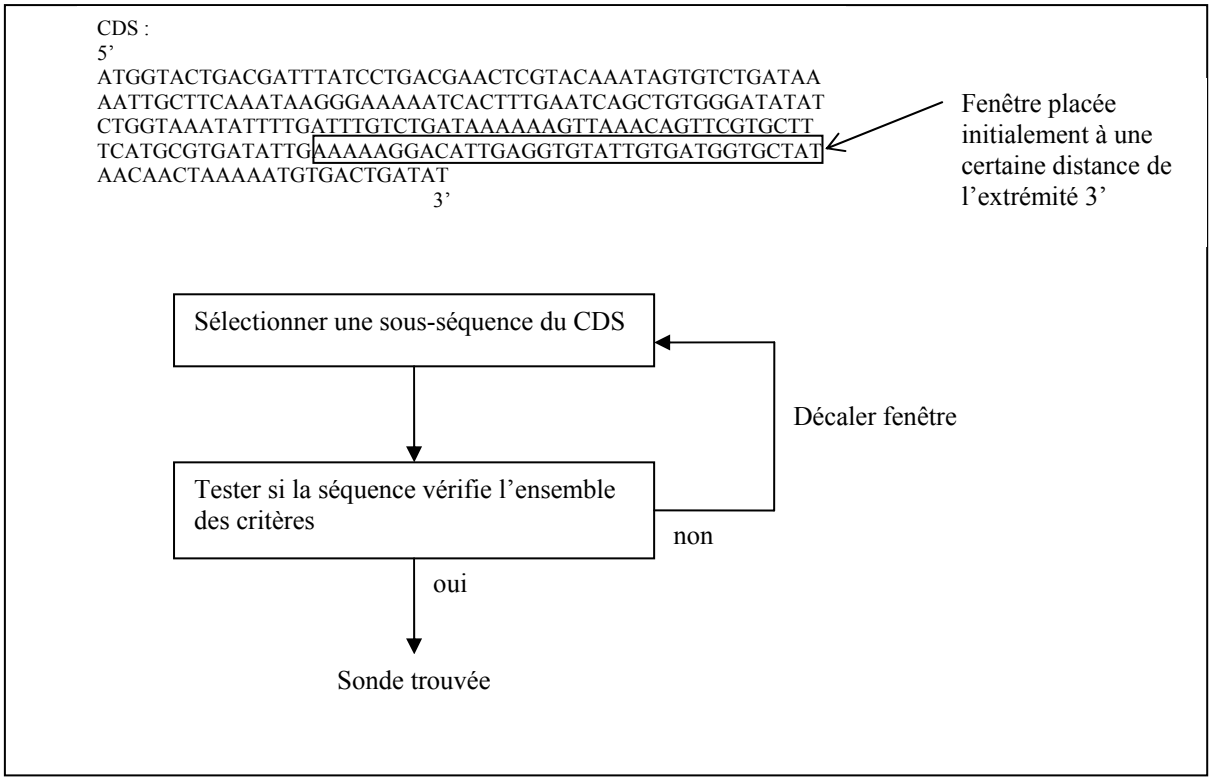


Figure 15 : Algorithme de base pour la recherche d'oligonucléotides.

Si l'on souhaite déterminer une sonde de longueur *l* la plus proche possible de l'extrémité 3' du CDS, il suffit de démarrer l'algorithme avec la sous séquence de longueur *l* situé à la fin du CDS et de choisir un décalage vers le début de la séquence. De plus, si l'on souhaite tester toutes les sondes possibles, il faut choisir un pas de décalage égal à une base.

L'autre façon de considérer les différents critères est de définir pour chacun d'eux une fonction de score :

$$f_i : \textit{Ensemble des séquences de longueur finie} \rightarrow \Re$$
$$s \mapsto f_i(s)$$
$$i \in \{\textit{critères}\}$$

Chaque fonction f_i associe à une séquence s un réel $f_i(s)$ tel que plus ce réel est élevé, meilleure est la séquence par rapport au critère i. Le score global d'une séquence

d'oligonucléotide est alors égal à la somme pondérée (suivant l'importance que l'on souhaite donner à chaque critère) des $f_i(s)$.

$$f(s) = \sum_{i \in \{critères\}} a_i f_i(s)$$

L'algorithme de recherche d'oligonucléotide consiste alors à évaluer la fonction de score f sur l'ensemble du CDS et à choisir la sonde dans la zone où f est la plus élevée [Nielsen et al 2003].

Quel que soit l'approche retenue, d'un point de vue informatique, il est nécessaire d'implémenter la vérification des différents critères. Pour certain d'entre eux, le calcul est trivial (température de fusion, composition en bases) alors que pour les autres il est nécessaire de mettre en place des algorithmes relativement complexes. Nous détaillons ces derniers dans les paragraphes suivants.

2.2 Recherche de la spécificité d'une séquence

Pour évaluer la spécificité d'une séquence, il est nécessaire de la comparer à l'ensemble des séquences susceptibles d'être présentes dans le mélange cible, et de tester si le critère de Kane est vérifié (voir 1.2). Ce problème peut être ramené au problème d'algorithmique appelé « Approximate String Matching » (ASM) :

> Etant données deux chaînes de caractères S et q, et une marge d'erreur k,
> Trouver toutes les sous chaînes s de S telles que : $d(s,q) \leq k$
>
> où d(s,q) est le nombre minimum d'insertions, délétions ou substitutions nécessaires pour que les deux chaînes soient identiques.

Dans notre contexte, S est l'ensemble des séquences du mélange cible et peut être de l'ordre de plusieurs centaines de mégabases. s représente la séquence de l'oligonucléotide à tester, soit quelques dizaines de bases. Pour tester la première condition du critère de Kane, k vaudra 0.75*longueur(s).

De très nombreux algorithmes ont été proposés pour résoudre ce problème [Navarro 2001].

Des algorithmes simples de programmation dynamique peuvent résoudre le problème avec une complexité O(|S|.|q|) [Sankoff et Kruskal 1983], mais leurs performances ne sont pas satisfaisantes pour notre problématique, lorsque |S| est de l'ordre de centaines de mégabases. Il est également possible de construire un automate fini déterministe pour résoudre le problème [Ukkonen 1985], ce qui donne une recherche en O(|S|) après un prétraitement pour construire l'automate. Des améliorations ont été apportées, notamment en utilisant la technique des « vecteurs de bits » [Wu and Manber 1992], ce qui ramène la complexité à O(E(|q|/w) . |S|), où w est la taille du mot machine, et E la fonction partie entière supérieure.

Enfin, il est possible d'utiliser des techniques d'indexation de texte pour résoudre le problème de « l'approximate string matching », même si beaucoup d'entre elles sont peu adaptées. En effet, on ne peut en principe pas séparer la chaîne d'ADN S en mots comme pour un texte normal. Les deux approches les plus utilisées sont les arbres des suffixes [Weiner 1973,

Ukkonen 1995], ainsi que les tableaux des suffixes [Manber and Myers 1993]. Ces structures sont construites à partir de S, elles indexent tous les mots et contiennent tous les suffixes de S.

Pour une séquence S = $lettre_0 lettre_1 ... lettre_n$ avec n un entier non nul, on appelle suffixe de S tout mot = $lettre_j lettre_{j+1} ... lettre_n$ avec j un entier compris entre 1 et n.

Considérons une chaîne S, contenant m caractères, l'arbre des suffixes pour cette chaîne est un arbre ayant exactement m feuilles, numérotées de 1 à m. Les arcs sont étiquetés par des sous chaînes de S. Tous les noeuds de l'arbre autre que le noeud racine, possèdent au moins deux noeuds fils. Les arcs définis des noeuds père vers les noeuds fils ne peuvent posséder des étiquettes commençant par le même caractère. Pour toute feuille i, la concaténation des étiquettes des arcs rencontrés de la racine de l'arbre à cette feuille, correspond exactement au suffixe S[i..m]. L'arbre des suffixes est construit en insérant successivement tous les suffixes du texte, avec une complexité $O(|S|^2)$. Cette complexité peut être réduite à $O(|S|)$ [Gusfield 1997]. La recherche d'une séquence exacte q dans la structure indexée se fait alors en $O(|q|)$. Malheureusement cette structure s'avère être de taille très importante en mémoire lorsque |S| est grand, et il est alors nécessaire d'utiliser des algorithmes construisant l'arbre en mémoire externe.

Un tableau des suffixes pour S est un tableau de tous les suffixes de S trié en ordre lexicographique. C'est une structure intimement liée à l'arbre des suffixes, en effet un nœud de l'arbre correspond à un intervalle dans le tableau. L'avantage par rapport à un arbre des suffixes est que l'espace mémoire occupé est beaucoup moins important. Cette structure a été utilisée dans un logiciel de recherche d'oligonucléotides pour puces à ADN [Li and Stormo 2001].

Cependant, les méthodes les plus utilisées en bioinformatique sont celles qui commencent par sélectionner dans S, des séquences « candidates » où l'on pourrait potentiellement retrouver la séquence q. Ensuite, ces candidats sont testés par des algorithmes de programmation dynamique. Les algorithmes BLAST [Altschul et al 1990] et FASTA [Lipman and Pearson 1985, Pearson and Lipman 1988] utilisent cette technique, et sont incontestablement les plus employés, à tort ou à raison, dans 90% des problèmes de bioinformatique : recherche de similitudes entre 2 séquences biologiques, recherche d'une molécule nouvellement séquencée dans les banques de données existantes, recherche de motifs…

BLAST est un algorithme recherchant les meilleures régions de similarités locales entre une séquence requête et les séquences d'une banque. Contrairement aux algorithmes vus précédemment, il utilise une méthode heuristique, il est donc très rapide mais sa sensibilité est parfois insuffisante. Des tests statistiques permettent de décider si l'alignement obtenu est significatif ou non, et les résultats fournis sont classés par ordre de signification. L'idée sous-jacente à l'algorithme est que les bons alignements doivent contenir quelque part des petits segments strictement identiques. Ces éléments constituent les points d'ancrage à partir desquels l'alignement est étendu. BLAST2 est une version de Blast qui autorise les insertions et les délétions alors que Psi-Blast est une version qui construit des matrices de substitution à partir d'alignements itératifs [Altschul et al 1997].

L'algorithme se déroule en trois phases :

- Extraction de tous les mots d'une taille fixée (W) contenus dans la séquence envoyée. Par défaut :

  W = 11 pour ADN

W = 3 pour protéines

- Recherche exacte de ces mots de taille W dans chaque séquence de la banque de séquences
- Blast essaye d'étendre la recherche de similarité. L'extension se fait aux deux extrémités des mots trouvés, de façon à trouver les régions de similitude les plus longues possibles ayant un score supérieur ou égal à un score seuil S.

Les alignements locaux trouvés sont appelés HSP (High-scoring Segment Pair).

Blast est l'algorithme le plus utilisé par les logiciels de conceptions d'oligonucléotides pour puces à ADN existants. Néanmoins, il est nécessaire de régler précisément les paramètres afin de vérifier le critère de Kane. Tout d'abord, il faut augmenter le seuil d'e-value maximum pour un HSP, afin d'être sûr que le résultat contient toutes les séquences de la banque susceptibles de ne pas vérifier le critère de Kane [Rouillard et al 2003]. De plus, il est souvent possible de réduire la recherche à l'un des deux brins selon le type de puce, alors que par défaut, Blast recherche les similarités à la fois sur les brins sens et anti-sens. Dans une expérience classique de puce à ADN, les séquences ADNc du mélange cible seront identiques au brin non codant de l'ADN génomique. Enfin, il est indispensable de réduire la taille W du mot utilisé dans la première phase de l'algorithme. En effet, on travaille avec des séquences requêtes courtes, et pour obtenir toutes les séquences de la banque contenant 15 bases consécutives identiques à la séquence requête, le paramètre W doit être réglé à la valeur minimale acceptable, soit 7 bases.

2.3 Calcul de la structure secondaire d'une séquence nucléique

Les méthodes de calcul de la structure secondaire d'une séquence nucléique sont essentiellement de deux types : minimisation d'énergie pour les séquences simples, et comparaison de séquences si l'on veut déterminer la structure de plusieurs séquences homologues. Seule la première technique est adaptée à la problématique de la conception d'oligonucléotides pour puces à ADN.

L'idée de l'approche par minimisation d'énergie est, comme son nom l'indique, de déterminer la structure correspondant à l'énergie libre minimale. En effet, à chaque configuration de la molécule, correspond une quantité d'énergie libre, et la configuration la plus stable est celle qui minimise l'énergie libre. De plus, en se repliant, la molécule adopte la configuration tridimensionnelle la plus stable.

L'algorithme de base pour le calcul de la structure secondaire d'énergie libre minimale est construit de la façon suivante :

On assigne une énergie libre à chaque appariement possible de bases, ces énergies étant déterminées par des calculs thermodynamiques [SantaLucia 1998] :

$e(r_i,r_j)$: donne l'énergie correspondant à l'appariement (i,j)

Exemple de valeurs données à 37°C :
GC -3 kcal/mol,

AT -2 kcal/mol,
GT -1 kcal/mol. (appariement possible mais avec une énergie de liaison beaucoup plus faible que les paires AT et GC présentent dans la structure de l'ADN double brin[6])

L'énergie totale d'une structure secondaire S sera :

$$E(S) = \sum_{i,j \in S} e(r_i, r_j)$$

L'idée est alors de trouver la structure S qui minimise E lorsque S parcourt l'ensemble des structures secondaires possibles de la séquence.

La technique utilisée est un algorithme classique de programmation dynamique. Soit E_{ij} l'énergie libre de la sous séquence $r_i, \ldots, r_j$. On calcule les $E_{i,j}$ pour tous les fragments possibles $r_i, \ldots, r_j$ de la séquence ARN, grâce à la formule :

$$E_{ij} = \begin{cases} 0 \quad \text{si } j - i < 4 \\ \min \left\{ \begin{array}{l} E_{i+1,j}, E_{i,j-1}, e(i,j) + E_{i+1,j-1}, \quad \min_{k=i+1}^{j-1} (E_{i,k} + E_{k+1,j}) \end{array} \right\} \quad \text{sinon} \end{cases}$$

En effet, ou bien :

- Les fragments de longueurs ≤ 4 ont une énergie nulle (pas d'appariement possible)
- r_i n'est pas apparié
- r_j n'est pas apparié
- r_i et r_j sont appariés ensembles
- r_i et r_j sont appariés mais pas ensemble. Dans ce cas, r_i est apparié avec r_{k1} et r_j avec r_{k2}, tels que $i < k1 < k2 < j$.

Pour n bases, deux matrices carrées de côté n sont crées, l'une stockant les énergies E_{ij}, l'autre contient le devenir de r_j. La table est ensuite "remontée" afin de construire la structure optimale.

Cet algorithme présente deux inconvénients majeurs : d'une part le fait d'assigner une énergie libre aux paires de bases ne reflète pas la réalité, car l'énergie dépend également des boucles qui sont formées par la molécule et pas uniquement des appariements de bases. D'autre part, il calcule uniquement la ou les structures d'énergie minimale, mais les imprécisions dans les calculs des paramètres d'énergie libre font qu'il peut être nécessaire de calculer d'autres structures secondaires d'énergie légèrement supérieure (repliement « sous-optimal »). Des améliorations ont été proposées dans ce sens [Williams and Tinoco 1986].

[6] Voir Annexe

Aujourd'hui, le logiciel le plus utilisé et qui fait référence dans le domaine est MFold [Zucker et al 1999]. Il implémente les avancées les plus récentes dans le domaine des algorithmes utilisant la minimisation d'énergie.

3 Les principaux logiciels de conceptions d'oligonucléotides pour puces à ADN

3.1 Généralités

Il y a encore un an ou deux, il existait extrêmement peu de logiciels de conception d'oligonucléotides pour puce à ADN. Actuellement, avec l'explosion de l'utilisation de la technique des biopuces en biologie moléculaire, les logiciels de ce type se multiplient et de nombreux laboratoires développent leur propre programme et le mettent à la disposition de la communauté bioinformatique. Tous les logiciels fonctionnent à peu de chose près selon le même principe : l'utilisateur fournit une liste de séquences nucléiques (CDS, ORF[7], …) pour lesquelles il souhaite déterminer un ou plusieurs oligos, ainsi qu'une base de données contenant en principe l'ensemble des transcrits potentiellement présents dans le mélange cible, afin de vérifier la spécificité de l'oligonucléotide. Il définit ensuite un ensemble de critères que devront satisfaire les oligos calculés (Tm, composition en GC…). Le logiciel fournit en sortie une liste de séquences accompagnées des paramètres correspondants.

D'un point de vue méthodologique, les logiciels diffèrent essentiellement par les critères choisis pour sélectionner les oligos : certains vont prendre en compte la possibilité pour un oligonucléotide de posséder une structure secondaire stable dans certaines conditions, d'autres vont permettre d'exclure certains motifs déterminés par l'utilisateur, comme la répétition d'une base unique. Aucun d'entre eux ne regroupe l'ensemble des critères.

D'un point de vue informatique, on peut séparer les logiciels de conception d'oligonucléotide en deux groupes : les logiciels de type client/serveur et les logiciels autonomes. Dans le cas d'une application de type client/serveur, l'utilisateur emploie un petit logiciel client pour envoyer les données à un serveur, généralement situé dans le laboratoire qui a conçu le logiciel. Ce serveur va effectuer le calcul des oligos et envoyer le résultat au client. Les avantages de ce type d'applications sont que l'utilisateur n'a pas besoin de grosses ressources de calcul à sa disposition et que les logiciels clients sont très faciles à installer et à utiliser. Cependant, la réalisation du calcul est dépendante de la disponibilité du serveur. Il est parfois possible de récupérer le logiciel serveur pour l'installer sur une machine dédiée, on revient alors à la problématique d'un logiciel autonome. Un autre inconvénient majeur de ce type d'application est que la base de données utilisée pour tester la spécificité des oligos doit être présente sur le serveur, ce qui limite les possibilités. En effet, on trouve en général sur ces serveurs uniquement les principaux génomes étudiés en biologie, ceux des organismes

[7] Open Reading Frame : région de l'ADN qui sépare deux codons STOP (donc potentiellement codante).

modèles. Si l'utilisateur réalise des expériences de puces à ADN sur des organismes peu étudiés, il ne pourra pas utiliser de tels logiciels ou devra contacter l'administrateur du serveur pour qu'il y ajoute une base de données.

Dans le cas d'un logiciel autonome, l'utilisateur maîtrise complètement la configuration des paramètres de la conception des oligos, et n'est pas dépendant d'une machine distante. Cependant, les applications de ce type sont plus difficiles à installer, nécessitent une certaine puissance de calcul et ont parfois besoin d'autres logiciels pour fonctionner (BLAST, MFold…).

3.2 Les logiciels de type Client/Serveur

3.2.1 OligoWiz

OligoWiz [Nielsen et al 2003] se compose d'un programme client écrit en Java servant à envoyer les données et à visualiser les résultats, et d'un serveur écrit en Perl fonctionnant sur une machine Silicon Graphics (SGI) Unix. Le programme client est téléchargeable gratuitement et permet de soumettre ses calculs au serveur du CBS (Center for Biological Sequence analysis) de l'université du Danemark. Il est également possible de se procurer le logiciel serveur, mais il est payant.

La méthode de recherche d'oligo est celle des fonctions de score (voir paragraphe 2.1) : pour chaque oligonucléotide potentiel, le programme prend en compte 5 paramètres et pour chacun de ces paramètres, il calcule un score pour l'oligo. Il fait ensuite une somme pondérée (par des coefficients définis par l'utilisateur) des scores, et renvoie l'oligo ayant obtenu le meilleur score. L'originalité du logiciel est que l'utilisateur peut également visualiser graphiquement les fonctions de score de tous les oligos potentiels et extraire d'autres oligos. Les critères pris en compte pour la recherche sont : la spécificité, la température de fusion (Tm), la position dans le transcrit, la complexité de la séquence et sa composition en bases autres que ATCG.

- Spécificité :

Pour un oligo donné, OligoWiz calcule un « score d'homologie » en utilisant le programme BLAST :

$$score = \frac{100 * L - \sum_{i=1}^{L} \max(h_{1i},...,h_{mi})}{100 * L}$$

où L est la longueur de l'oligo, m le nombre de hits BLAST de l'oligo en position i et $\{h_{1i},\ldots,h_{mi}\}$ les hits BLAST.

Un oligo ne présentant aucune homologie avec une séquence de la base de donnée aura un score de 1, alors qu'un oligo présentant 100% d'homologie tout le long de sa séquence aura un score de 0.

Le logiciel permet de ne pas considérer les homologies inférieures à un certains seuil (par défaut 70%, proche du critère de Kane) et inférieure à une certaine longueur (par défaut 15bp,

critère de Kane). Mais parmi toutes les homologies considérées, il ne définit pas de seuil au-delà duquel l'oligo ne serait plus spécifique. Il renverra simplement l'oligo ayant le score le plus élevé.

- Tm :

OligoWiz utilise le modèle du plus proche voisin pour l'estimation de la Tm. Il calcule la température de fusion moyenne O_{Tm} de tous les oligos potentiels de toutes les séquences fournies en entrée et définit un score pour ce critère :

$$Tm\ score = \left| Tm - O_{Tm} \right|$$

- Position dans le transcrit :

OligoWiz calcule un score de position de façon à ce que plus l'oligo est proche de l'extrémité 3' de la séquence fournie en entrée, plus son score est élevé.

- composition en bases autres que ATCG :

OligoWiz privilégie les oligos dont la séquence contient uniquement les lettres A, T, G ou C, par rapport aux oligos contenant des bases ambiguës (par exemple N qui peut remplacer n'importe quelle base). Il attribue un score de 1 aux premières et un score de 0 aux secondes.

URL : http://www.cbs.dtu.dk/services/OligoWiz/

3.2.2 ROSO

ROSO [Reymond et al 2004] est un programme C qui utilise BLAST pour le calcul de spécificité. Il est installé sur un serveur du Pôle BioInformatique Lyonnais et est accessible via un site web. Il est également possible de récupérer le code source et les exécutables pour les systèmes les plus courants sur demande auprès des auteurs.

Le programme accepte deux fichiers en entrée : les CDS des gènes à spotter sur la puce ainsi qu'un fichier contenant des séquences susceptibles de se trouver dans le mélange cible mais non spottées. Ce deuxième fichier sera utilisé pour le test de spécificité. Il est possible de spécifier un grand nombre de paramètres : nombres de sondes calculées par gène (avec recouvrement ou non), brin utilisé pour la recherche (sens ou antisens), concentration ionique de la solution, plage de Tm souhaitée, paramètres de structure secondaire, etc… L'algorithme de recherche d'oligonucléotides se compose de 5 étapes :

1) Filtrage des séquences fournies en entrée (élimination des gènes identiques, des répétitions de bases…). La recherche des sondes peut également se faire uniquement dans une certaine zone des CDS si l'utilisateur le spécifie.

2) Recherche des hybridations croisées potentielles à l'aide de BLAST. Les paramètres sont définis de façon à détecter au moins les identités de 70% sur 20 bases. Un score d'hybridation croisée est calculé pour chaque oligo potentiel.

3) Elimination des oligos possédant une structure secondaire stable.

4) Calcul de la Tm de chaque oligo candidat en utilisant le modèle thermodynamique du plus proche voisin et sélection d'un ensemble d'oligo (au minimum un par gène) qui minimise la variabilité de la Tm.

5) Sélection de l'ensemble final d'oligo sur 4 critères (composition en GC, première et dernière base, répétitions, énergie libre).

L'originalité de l'algorithme est qu'il sépare la recherche de spécificité de l'étape de recherche des sondes proprement dite. En effet, le programme effectue un BLAST des gènes entiers entres eux plutôt que de le faire pour chaque sonde potentielle. Ceci permet en plus, une fois le calcul des scores de spécificité effectué, de réaliser une recherche de sondes itérative afin d'assurer une uniformité des paramètres thermodynamiques, sans avoir à recalculer la spécificité.

URL : http://pbil.univ-lyon1.fr/roso/

3.2.3 Autres logiciels

OligoDesign [Tolstrup et al 2003] est un logiciel accessible via un « frontal » web. Ce dernier permet de lancer un ensemble de programmes écrit en C et en Perl. Sa grande particularité est qu'il permet d'effectuer un design d'oligonucléotides contenant des LNA (Locked Nucleic Acid). Un LNA est un nucléotide modifié qui peut se substituer aux nucléotides classiques dans une séquence et qui permet d'augmenter la spécificité et la sensibilité dans les expériences reposant sur l'hybridation.

URL : http://oligo.lnatools.com/expression/

Osprey [Gordon and Sensen 2004] est un logiciel écrit en C accessible sur internet via un CGI écrit en Perl. Il permet d'effectuer une recherche de sondes pour puces à ADN, mais également d'amorces PCR, et d'amorces pour le séquençage de l'ADN. Il implémente une méthode originale de recherche de spécificité pour les oligos (« Position-Specific Scoring Matrices ») qui tire parti de cartes d'accélération[8] commercialisées spécialement pour les calculs de bioinformatique.

URL : http://osprey.ucalgary.ca/

3.3 Les logiciels autonomes

3.3.1 OligoArray 1.0

OligoArray [Rouillard et al 2002] est un logiciel écrit en Java distribué gratuitement et dont le code source est disponible sous licence GPL. Il nécessite le programme BLAST pour

[8] cartes commercialisées par la société TimeLogic avec des architectures spécialisées pour l'exécution d'algorithmes tels que BLAST, Smith & Waterman, ou les modèles de Markov cachés

fonctionner. Pour le calcul de la structure secondaire de l'oligonucléotide, OligoArray utilise le programme MFold. Il est nécessaire d'avoir une connexion à Internet pour effectuer ce calcul sur le serveur de MFold, ce qui d'une certaine façon pourrait placer OligoArray dans la catégorie des programmes Client/Serveur. Mais la quasi-totalité des calculs est effectuée sur la machine locale. Il existe également une version totalement autonome si l'on dispose du logiciel MFold installé en local mais elle apparaît plus instable (arrêts intempestifs du programme).

Les critères pris en compte pour la recherche d'oligonucléotide sont : la spécificité, la position dans le transcrit, la température de fusion, la structure secondaire de l'oligo, et la présence de certains motifs dans la séquence de l'oligo.

La stratégie adoptée par OligoArray est la suivante : étant donnée une séquence fournie en entrée, il extrait une sous séquence constituée des *n* dernières bases (*n* étant la longueur souhaitée pour les oligos) et vérifie l'ensemble des critères cités plus haut. Si cette sous séquence vérifie tous les critères, elle constituera un oligonucléotide satisfaisant, et le programme passe au gène suivant. Dans le cas contraire, le programme effectue un décalage de 10 bases vers l'extrémité 5', extrait de nouveau une sous séquence et vérifie les critères. Le processus continue jusqu'à ce qu'un oligo satisfaisant soit trouvé. Si aucun n'est trouvé, il fait un nouveau balayage de la séquence avec un critère de spécificité moins strict. Si à nouveau aucun oligo n'est trouvé, le programme renvoie l'oligo présentant le moins d'hybridation croisée potentielle. Pour l'ensemble des critères considérés, le programme renvoie une réponse de type oui/non : si l'oligo vérifie le critère, il est conservé, sinon il est rejeté.

Détaillons les deux principaux critères utilisés :

- Spécificité :

Le logiciel analyse chaque hit BLAST de l'oligo contre la base de donnée et effectue le test :

Si ((l $\geq$ 36 et l < 51 et p $\geq$ 60)
ou
(l $\geq$ 15 et l < 36 et p $\geq$ 70))

Alors rejeter la séquence

Sinon conserver la séquence (elle est spécifique)

où l désigne la longueur du hit, et p le pourcentage d'identité.

- Position dans le transcrit :

L'oligo renvoyé est la séquence la plus proche de l'extrémité 3' vérifiant tous les autres critères. Il est également possible de définir une limite pour arrêter la recherche au-delà d'une certaine distance à cette extrémité.

URL : http://berry.engin.umich.edu/oligoarray/

3.3.2 ProbeSelect

ProbeSelect [Li and Stormo 2001] est un logiciel écrit en C++ développé sous Sun Solaris, dont le code est portable sous Linux et éventuellement sous Windows. Il ne nécessite aucun autre programme pour fonctionner. Il s'est cependant révélé difficile à installer et à utiliser (erreurs à la compilation, documentation sommaire).

L'approche retenue pour la recherche d'oligonucléotides est relativement complexe et est centrée sur le problème de la spécificité des séquences. L'algorithme se compose de 7 étapes :

1) Construction d'un tableau des suffixes pour l'ensemble des séquences codantes du génome de l'organisme considéré.

2) Construction d'un « landscape » pour chaque gène fournit en entrée.

3) Ces structures sont utilisées pour déterminer une liste d'oligos candidats (10 à 20) pour chaque gène (séquences qui minimisent la somme des fréquences de leurs sous mots dans l'ensemble du génome).

4) Pour chaque oligo candidat, recherche des hybridations croisées potentielles

5) Localisation de ces hybridations croisées à l'intérieur des gènes

6) Calcul de l'énergie libre et de la Tm pour chaque hybridation

7) Sélection des oligos qui ont les conditions d'hybridation les plus stables avec leur cible et qui permettent une bonne discrimination des autres cibles potentielles

Les critères considérés pour obtenir un oligonucléotide satisfaisant ne sont donc pas évalués séparément, mais sont imbriqués dans l'algorithme.

- Spécificité :

Il y a une hybridation croisée potentielle si l'on trouve dans le génome une séquence (autre que le gène cible) identique à l'oligo avec un certain nombre de mismatches autorisés :

4 mismatches pour les oligos longs de 20 à 25 bases
10 mismatches pour les oligos longs de 50 bases
20 mismatches pour les oligos longs de 70 bases

Le logiciel est disponible sur demande auprès des auteurs.

3.3.3 Autres logiciels

ProMide [Rahmann 2003] est un ensemble de script Perl qui utilise la plus longue sous-séquence commune comme mesure de spécificité des oligos. Il utilise des structures de données complexes (« enhanced suffix array ») et certaines propriétés statistiques des séquences.

URL : http://oligos.molgen.mpg.de/

Oliz [Chen and Sharp 2002] est également un ensemble de script Perl qui utilise une approche assez classique, avec utilisation de BLAST pour le test de spécificité. Il nécessite cependant un certain nombre de logiciels pour fonctionner (cap3, clustalw, EMBOSS prima), ainsi qu'une base de donnée au format UniGene. L'originalité de la méthode utilisée vient du fait que les oligos sont recherchés dans la région 3'UTR des gènes, région très spécifique et dont les séquences sont largement disponibles dans les banques d'ESTs (Expressed Sequence Tag).

URL : http://www.utmem.edu/pharmacology/otherlinks/oliz.html

OligoArray 2.0 [Rouillard et al 2003] est l'évolution du logiciel étudié plus haut. D'un point de vue informatique, il possède les mêmes caractéristiques (programme JAVA utilisant BLAST et MFold) mais il adopte une approche assez différente dans la méthode de sélection des oligos. Il utilise une approche thermodynamique pour calculer la spécificité des oligos.

URL : http://berry.engin.umich.edu/oligoarray2/

Featurama est un logiciel écrit en C++ qui utilise une approche classique, proche d'OligoArray 1.0. Les critères pris en compte sont la spécificité, la température de fusion, la composition de la séquence, la répétition de bases.

URL : http://probepicker.sourceforge.net/

OligoPicker [Wang and Seed 2003] est un programme Perl qui utilise également une approche classique. Les critères pris en compte sont la spécificité (utilisation de BLAST), la température de fusion, la position dans le transcrit.

URL : http://pga.mgh.harvard.edu/oligopicker/

DEODAS est un logiciel écrit en C, C++ et Python, qui intègre plusieurs programmes existants (ClustalW, EMBOSS, BLIMPS) au sein d'une chaîne de traitements. L'originalité est qu'il permet de rechercher des oligonucléotides dégénérés : les séquences de ces sondes peuvent comporter des bases autres que les quatre bases classiques, appelées bases dégénérées qui remplacent plusieurs bases classiques. Ainsi, les sondes déterminées peuvent reconnaître des familles de gènes plutôt qu'un seul gène spécifique.

3.4 Comparaisons et Tests

Les tableaux 1 et 2 présentent une comparaison des principaux logiciels de conception d'oligonucléotides pour puces à ADN respectivement d'un point de vue méthodologique

(algorithmes et critères utilisés pour sélectionner les oligos) et d'un point de vue logiciel (langage, autres logiciels nécessaires…).

Le tableau 2 présente également une comparaison en terme de temps d'exécution. Il est difficile de comparer les résultats des logiciels. En effet, pour un même gène, deux programmes peuvent fournir des oligos différents, mais ces derniers peuvent être tout à fait satisfaisants et compatibles avec les critères fournis par l'utilisateur. Seuls deux logiciels considérant exactement les mêmes critères pour leur recherche et proposant les mêmes options à l'utilisateur devraient fournir les mêmes résultats.

On peut cependant comparer les temps d'exécution des différents logiciels. Pour cela, nous avons choisi de mesurer le temps mis par chaque logiciel pour calculer des oligonucléotides pour tous les CDS du chromosome I de *Saccharomyces cerevisiae*.

Dans le cas des logiciels autonomes, la base de donnée utilisée pour vérifier la spécificité des séquences est constituée par l'ensemble des séquences codantes du génome de *Saccharomyces cerevisiae*, disponible sur le site Saccharomyces Genome Database (http://www.yeastgenome.org/)[9] :

Les séquences fournies en entrée et pour lesquelles on veut déterminer des oligos sont les CDS du chromosome I extraites de ce même fichier. On souhaite obtenir un oligo par CDS.

Les tests ont été effectués sur un Pentium IV 1,8 GHz avec 256 Mo de RAM.

[9] ftp.stanford.edu/pub/yeast/data_download/sequence/genomic_sequence/orf_dna/orf_coding.fasta.gz : 5886 séquences, 8 775 285 bps.

	Méthode de recherche d'oligos								
	Critères utilisés							Algorithme de recherche	Particularités
	Hybridation croisée	Tm	Position	"Complexité" de la séquence	Composition en bases	Séquence interdite	Structure secondaire		
OligoWiz	●	●	●	●	●			Pour chaque oligo potentiel et pour chaque critère, calcul d'un score indépendant. L'oligo renvoyé est celui qui possède le plus haut score cumulé.	Le logiciel calcule un seul oligo par gène. Mais il y a possibilité de définir d'autres oligos en visualisant graphiquement les fonctions de score.
ROSO	●	●	●		●	●	●	A partir de l'ensemble des oligos potentiels, éliminations des oligos ne répondant pas aux critères définis, pris successivement.	Séparation de la recherche de spécificité (effectuée sur les gènes entiers) de la phase de sélection de sonde. Possibilité de recherche itérative sans relancer le test de spécificité.
OligoArray 1.0 (client / serveur)	●	●	●			●	●	Balayage du gène à partir de l'extrémité 3'. Si la sous séquence satisfait les critères définis, elle est conservée, sinon décalage de 10 bases.	Le logiciel ne permet de calculer qu'un seul oligo par gène.
OligoArray 1.0 (standalone)	●	●	●			●	●	Balayage du gène à partir de l'extrémité 3'. Si la sous séquence satisfait les critères définis, elle est conservée, sinon décalage de 10 bases.	Le logiciel ne permet de calculer qu'un seul oligo par gène.
ProbeSelect	●				?		?	assez complexe. Utilisation de "suffix array" pour la recherche de spécificité	Ne nécessite aucun autre programme pour fonctionner. Documentation sommaire
Oliz	●	●			●			approche classique par balayage avec utilisation de BLAST pour le test de spécificité	les oligos sont recherchés dans la région 3'UTR des gènes, région très spécifique et dont les séquences sont largement disponibles (EST).

Tableau 1: Comparaison des principaux logiciels de design d'oligonucléotides pour puces à ADN en terme de critères utilisés pour sélectionner les oligos.

	Informatique							
	type d'application	Particularités	OS	langage	open source	licence	autres logiciels utilisés	tests[10]
OligoWiz	Client / Serveur	Accès au serveur via un client léger en Java	client : Tous serveur : SGI Unix	client : Java serveur : Perl	Non	client : Gratuit serveur : payant	serveur :BLAST, saco_patterns	2min 22s
ROSO	Client / Serveur	Accès au serveur via un frontal web	serveur : tous	serveur : C	Oui	Gratuit	serveur : BLAST	25min
OligoArray 1.0 (client / serveur)	Client / Serveur	Le serveur sert uniquement pour calculer la structure secondaire	client : Tous	client : Java	Oui	client : Gratuit	client : BLAST	57min
OligoArray 1.0 (standalone)	Autonome		Unix (à cause de Mfold)	Java	Oui	Gratuit	BLAST, Mfold	15min
ProbeSelect	Autonome		Unix	C++	Oui	Gratuit		n.c.
Oliz	Autonome	Nécessite la base de donnée de l'organisme étudié au format Unigene	Unix	Perl	Oui	Gratuit	BioPerl, BLAST, cap3, clustalw, EMBOSS prima	n.c.

Tableau 2 : Comparaison des programmes de design d'oligonucléotides pour puces à ADN du point de vue logiciel.

[10] Les temps correspondants aux logiciels client/serveur sont donnés à titre indicatif (moyenne de 10 soumissions identiques à différentes heures).

4 Bilan

La plupart des logiciels présentés ici ont été testés sur des données de références. Les résultats laissent apparaître que la principale difficulté dans la conception des oligonucléotides est le problème de la spécificité des sondes. Ainsi, au sein du génome de la levure *Saccharomyces cerevisiae*, 253 CDS (4,5% de l'ensemble des CDS) ne peuvent être représentées par une unique sonde spécifique [Talla et al 2003]. Même avec une version améliorée du logiciel OligoArray (2.0) présenté au paragraphe 3.3.1, basée sur une approche thermodynamique pour le calcul de spécificité, il est impossible de déterminer une sonde spécifique pour 7% des CDS d'*Arabidopsis thaliana* [Rouillard et al 2003]. Ce pourcentage est d'autant plus élevé que le modèle biologique est complexe.

L'analyse des logiciels de conception d'oligonucléotides existants nous a permis de soulever un autre problème, purement informatique celui-ci : ces logiciels sont difficilement modifiables en vue de les adapter à un modèle biologique particulier. En effet, même si leur code source est disponible, ils sont souvent programmés avec une approche fonctionnelle et peu documentés, il est donc difficile d'effectuer une rétro ingénierie en vue de leur modification. C'est pourquoi, avant de proposer de nouvelles méthodes de conception de sondes, il nous a semblé intéressant de nous pencher sur la problématique de la conception de logiciel dans le domaine de la « bioinformatique des puces à ADN ». Ceci nous permettra ensuite de proposer des solutions satisfaisantes sur le plan du Génie Logiciel.

Le chapitre suivant s'intéresse donc aux expériences de biopuces en général, mais cette fois vues sous l'angle du Génie Logiciel. Nous tentons de rechercher des solutions aux problèmes de réutilisabilité des composants logiciels soulevés ci-dessus.

Chapitre III

Les puces à ADN : aspects Génie Logiciel

1 Introduction

Nous avons vu au chapitre précédent que les logiciels de conception d'oligonucléotides existants utilisaient une approche de programmation qui les rendait difficilement réutilisables. De plus, leur faible documentation complique la rétro ingénierie nécessaire à leur adaptation à un modèle biologique particulier. Ce problème s'inscrit dans un cadre plus général : dans le domaine des logiciels pour puces à ADN, il existe un manque de modèles de composants logiciels réutilisables, ainsi qu'un manque de standardisation des données. Dans ce chapitre nous nous intéressons à l'utilisation des techniques actuelles de Génie Logiciel dans le domaine des puces à ADN et notamment aux standards que tente d'imposer le consortium MGED (Microarray Gene Expression Data).

Dans les deux premières parties, nous introduisons les notions de bases nécessaires à la compréhension des standards MGED : l'ingénierie des modèles, en nous limitant à la démarche MDA (Model Driven Architecture) pour la conception d'applications, ainsi que la notion d'ontologie en ingénierie des connaissances. La troisième partie présente en détails les standards MGED pour les puces à ADN. En conclusion, nous replaçons notre problématique de conception d'oligonucléotides dans le contexte des normes MGED et identifions les limites de ces normes dans leur capacité à modéliser notre problématique.

2 L'ingénierie des modèles

2.1 Origine

Le domaine des langages de programmation, des architectures middleware et des méthodes de génie logiciel est soumis à des effets de mode et à des évolutions constantes. Ainsi, dans le domaine des intergiciels (middlewares), Corba se voulait suffisamment puissant et ouvert pour répondre à tous les besoins et devenir le standard universel. Pourtant, il est aujourd'hui en perte de vitesse, face aux autres architectures de composants telles les EJBs (Entreprise Java Bean) ou .NET. Les entreprises ne veulent plus investir des sommes énormes dans des middlewares qui n'ont pas une pérennité assez élevée et sont demandeuses de nouvelles approches indépendantes d'un middleware particulier pour la création d'applications basées sur les composants.

Pour répondre à ce besoin, l'Object Management Group (OMG) a proposé la démarche de développement « MDA » (Model Driven Architecture) [OMG 2003-b]. Elle permet de séparer les spécifications fonctionnelles d'un système, des spécifications de son implémentation sur une plate-forme donnée. La mise en œuvre du MDA est entièrement basée sur les modèles et leurs transformations.

L'ingénierie des modèles, généralisation du MDA, est une approche qui met les modèles, et non pas les programmes, au centre de la démarche en Génie Logiciel [Favre 2004]. Les avantages sont nombreux : indépendance vis-à-vis des évolutions technologiques, meilleure maîtrise de la complexité, meilleure réutilisation, etc.

2.2 Architecture globale de MDA

MDA se découpe en quatre couches principales, ou étapes, qui se réfèrent chaque fois à des standards déjà adoptés par l'industrie ou en cours de normalisation (Figure 16).

La première étape, au cœur de MDA, utilise les technologies UML (Unified Modeling Language), MOF (MetaObject Facility) et CWM (Common Warehouse Metamodel) spécifiées par l'OMG pour modéliser la logique métier de l'application.

Ce modèle métier est ensuite spécialisé dans une technologie propre à un middleware particulier. On retrouve les standards actuels tels que les EJBs, Corba, .NET ou encore les WebServices.

L'anneau extérieur du cercle représente les services. Ceux-ci permettent par exemple de gérer les transactions, la persistance, ou encore les événements. Cette couche permet de prendre en compte des services présents au sein de plusieurs middlewares.

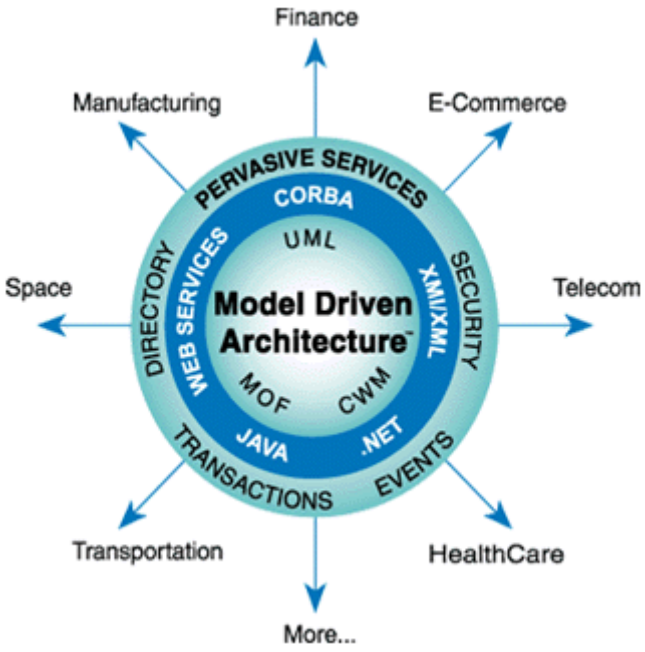


Figure 16 : Architecture globale de MDA (figure OMG).

Enfin, la dernière étape se situe à la périphérie du noyau. Elle se base sur les profils UML et permet de proposer des cadriciels (frameworks) spécifiques au domaine d'applications du logiciel (télécommunication, biologie, etc.).

La démarche MDA pour le développement d'une application peut se découper en quatre points :

La réalisation d'un modèle indépendant de toute plateforme appelé PIM pour « Platform Independant Model ».

L'enrichissement de ce modèle et le stéréotypage des classes.

Le choix d'une plateforme de mise en œuvre et la génération du modèle spécifique correspondant appelé PSM (« Platform Specific Model »).

Le raffinement de celui-ci jusqu'à l'obtention d'une implantation exécutable.

2.3 Le Platform Independant Model (PIM)

La première étape lors de la conception d'une application à l'aide de MDA est de se concentrer sur les fonctionnalités et les comportements métiers de l'application, qui sont totalement spécifiques à l'application, mais indépendants de la plate-forme utilisée pour la mettre en œuvre. C'est ce que l'on nomme la logique métier (business logic en anglais). Cette étape sera réalisée par un architecte spécialisé dans le domaine de l'application.

Concrètement, l'architecte met au point le diagramme de classes du domaine : c'est le Platform Independant Model (PIM).

Pour mettre au point la logique métier, trois standards sont utilisés :

- UML, pour concevoir, visualiser, spécifier et documenter les composants du métier.
- MOF pour méta-modéliser les composants, par l'intermédiaire de la norme d'échange XMI (XML Metadata Interchange).
- CWM pour modéliser les flux entre entrepôts de données et les processus de traitement d'informations.

Le PIM étant indépendant de toute plate-forme logicielle, il est ensuite nécessaire de le stéréotyper (phase de « marking » en Anglais) en vue de donner une sémantique aux classes du domaine. Les « marques » sont des extensions du modèle initial, non intrusives, qui spécifient les informations nécessaires pour la transformation de modèle sans polluer le modèle original. Elles sont définies par les modèles de marquage, lesquels décrivent la structure et la sémantique d'un ensemble de types de marques. Ces stéréotypes facilitent le passage automatique du PIM vers le PSM.

2.4 Le Platform Specific Model (PSM)

Une fois le PIM mis en place, le projet est confié à l'architecte technique. Il s'agit cette fois de traduire le modèle décrit dans les termes métiers en un modèle spécifique d'un middleware donné : le Platform Specific Model (PSM). Le PSM combine les spécifications métiers décrites dans le PIM avec des spécifications qui permettent d'utiliser ce PIM su la plate-forme cible. Le PSM est donc le diagramme de classes UML représentant les classes du modèle métier dans le contexte du middleware cible.

Le PSM est obtenu par transformation du PIM (Figure 17). Cette opération de transformation utilise les stéréotypes mis en place lors de l'élaboration du PIM stéréotypé : ils servent à paramétrer le PIM pour obtenir le PSM. MDA ne définit pas la méthode à utiliser pour transformer le PIM en PSM, cette opération relevant du domaine de l'implémentation. Ceci explique le rectangle blanc dans la figure.

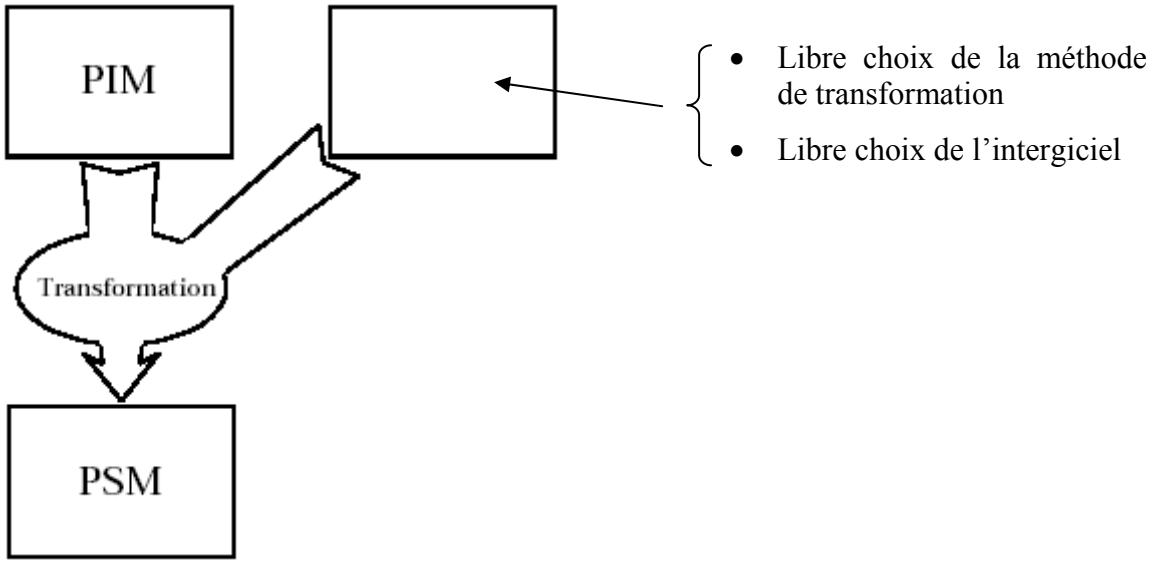


Figure 17 : Transformation du PIM pour obtenir le PSM [OMG 2003-b].

Un des objectifs de la démarche MDA est de permettre à terme, et pour certains domaines, la transformation automatique du PIM en PSM par les ateliers de génie logiciel. Actuellement, peu d'outils proposent effectivement cette transformation. On peut citer l'environnement de développement OptimalJ de Compuware[11] qui implémente la quasi-totalité des standards MDA mais qui se limite à la seule plateforme J2EE, Model-in-Action, une suite logiciel de Mia software[12] qui inclut la génération de code et la transformation de modèles, ou les projets open source AndroMDA[13] et OpenMDX[14].

[11] www.compuware.fr
[12] www.mia-software.com
[13] www.andromda.org
[14] www.openmdx.org

2.5 Utilisation de MDA

Aujourd'hui, MDA n'est plus simplement une théorie, plusieurs grandes entreprises proclament avoir implémenté une approche de cette démarche. De plus, des travaux ont démontré la viabilité de l'application de MDA aux services Web [Bézivin et al 2004], aux systèmes embarqués [Gokhale et al 2002, Gray et al 2004], ou aux systèmes d'information [Sims 2004].

Dans le domaine des sciences de la vie, l'OMG dispose d'un groupe de travail, le « Life Science Research (LSR) group » qui a pour but de faciliter la communication et l'interopérabilité entre les ressources informatiques de ce domaine [Benton 2000]. Bien que créé avant l'initiation de MDA, le LSR group en a adopté la démarche et produit aujourd'hui des Platform Independant Models (voir paragraphe 4.3).

Cependant, certaines zones d'ombres subsistent : l'approche MDA n'est pas encore complètement expérimentée et plusieurs problèmes ne sont pas encore résolus. Notamment, le langage de transformation autour de MDA n'est pas normalisé. Seul l'avenir pourra nous dire si MDA est seulement une mode, ou s'il remportera une adhésion massive et transformera la démarche du génie logiciel.

Mais MDA n'est en fait qu'une variante particulière de ce que l'on nomme l'Ingénierie Dirigée par les Modèles (IDM ou « Model Driven Engineering » en anglais). Et cette dernière est quasiment assurée d'influencer profondément les pratiques du génie logiciel dans les années à venir. En effet, les acteurs de ce domaine, ainsi que le monde académique, ont pris conscience qu'une partie importante du travail d'ingénierie a toujours été d'établir des modèles, des méta-modèles, et d'en dériver des outils. C'est pourquoi ils ont décidé de mettre les concepts de modèle et de méta-modèle au centre de l'approche MDE [Bézivin 2004]. Cette approche se généralise à tous les types de modèles et pas uniquement à ceux liés aux standards de l'OMG (MOF/UML). Enfin, des acteurs industriels majeurs comme Microsoft et IBM annoncent déjà des produits et un support fort aux approches MDE, mais avec une orientation vers les « Domain Specific Languages » (DSL) plus que vers MDA. Les DSL sont de petits langages spécifiques de domaines adaptés à des corporations particulières ou à des besoins particuliers. Ces langages de petites tailles sont facilement manipulables, transformables ou combinables. Donc au final, il y a de fortes chances que l'une ou l'autre des variantes de l'approche MDE révolutionne le monde du génie logiciel, même si ce n'est pas MDA.

2.6 Lien avec les ontologies

Nous avons vu qu'une partie du processus MDA consistait à réaliser un modèle du domaine de l'application, indépendamment de toute considération technologique. Or l'objectif des ontologies est précisément de permettre la compréhension des éléments d'un domaine, et les relations entre ces éléments. L'utilisation conjointe de l'approche MDA et des ontologies est donc pertinente, certains auteurs proposent même une unification des deux technologies, et présentent les ontologies comme pouvant être représentées par un sous ensemble de MDA [Atkinson 2004]. Le paragraphe suivant présente la notion d'ontologie en ingénierie des connaissances et plus particulièrement son utilisation en biologie.

3 Les ontologies en ingénierie des connaissances

3.1 Définition

Tout d'abord, il est nécessaire de préciser ce que l'on entend par ontologie. En effet, ce terme est employé dans de nombreux domaines, et sa définition reste assez vague. Du grec ontos (être) et logos (étude), littéralement « science de ce qui est », le terme remonte à Aristote pour ce qui est de son acception philosophique. Le lecteur intéressé pourra se reporter à [Guarino and Giaretta 1995] pour une étude des différentes définitions utilisées en Intelligence Artificielle.

Pour le consortium MGED, la notion d'ontologie est celle donnée par [Gruber 1993], et c'est aussi la définition la plus utilisée en ingénierie des connaissances. Elle se divise en trois points :

1. Une ontologie se réfère à un domaine.

2. Une ontologie est formée par des concepts et par des relations entre ceux-ci.

3. Une ontologie est une spécification d'une conceptualisation.

En d'autres termes, une ontologie est une description formelle explicite des concepts d'un domaine de la connaissance [Gruber 1995]. Elle comprend deux choses :

- un **vocabulaire contrôlé** commun aux experts qui ont besoin de partager une connaissance. Ce vocabulaire se présente sous la forme d'un catalogue sémantique, dont les descriptions sont à la fois concises, non ambiguës, et qui se doit d'être exploitable par un logiciel (description formelle) comme par un opérateur humain (description littéraire).
- Une représentation des **relations entre ces termes**, qui définissent cette connaissance. Ces relations sont souvent des relations de composition et d'héritage au sens des concepts du modèle objet.

3.2 Les ontologies en biologie

Depuis une quinzaine d'années, les projets de séquençage des génomes complets ainsi que les nouvelles techniques à haut débit telles que les puces à ADN produisent des masses de données très importantes que les biologistes ont beaucoup de mal à interpréter et à exploiter. De ce fait, l'annotation de ces données, afin de leur apporter un sens, devient un enjeu majeur de la biologie et par extension de la bioinformatique.

De plus, il n'est pas rare qu'un même terme ait une signification différente suivant les domaines, ou même parfois, ce qui est plus grave, entre les chercheurs d'une même communauté. L'exemple le plus frappant, dans le domaine des puces à ADN, est celui des termes utilisés pour désigner les séquences fixées sur le support solide et les séquences

marquées présentes dans le mélange d'hybridation. Dans les premières publications sur les puces à ADN, les termes « probe » (sonde) et « target » (cible) désignaient indifféremment l'un ou l'autre des types de séquences suivant les articles. Aujourd'hui, même s'il est clairement défini que le terme « sonde » désigne la séquence fixée sur le support solide et « cible » la séquence du mélange d'hybridation, le consortium MGED tente d'imposer, via son ontologie, deux nouveaux termes afin d'éviter toute confusion (nous en discuterons au paragraphe 4.4).

Les ontologies en biologie vont donc pouvoir être utilisées d'une part comme un dictionnaire de termes permettant de désigner les entités manipulées sans ambiguïté, et d'autre part pour décrire la structure des informations disponibles. Leur liste ne cesse de croître, elles sont répertoriées par le projet OBO (Open Biomedical Ontologies, http://obo.sourceforge.net).

Une des principales ontologies utilisée en biologie/bioinformatique est Gene Ontology [GO Consortium 2004]. Son objectif prioritaire est de fournir un vocabulaire contrôlé pour l'annotation des gènes.

4 Les travaux de la « MGED Society » et de l'OMG

4.1 Introduction

Crée en 1999, le consortium **MGED** (Microarray Gene Expression Data, www.mged.org) rassemble des universitaires biologistes et informaticiens ainsi que de grands groupes privés. Son but est de développer et de promouvoir l'utilisation de standards dans le domaine des expériences de puces à ADN. Il est composé de chercheurs issus des principaux centres de bioinformatique mondiaux (NCBI, EBI, DDBJ), du laboratoire de l'université de Stanford pionnier en matière de puces à ADN, et des principales sociétés privées du domaine (Affymetrix, Rosetta). Il dispose également de sponsors tels que IBM, Sun ou SAS.

Les travaux du consortium MGED sont divisés en trois grands axes [Stoeckert et al 2002] :

- **MIAME (Minimum Information About a Microarray Experiment) :** il s'agit de définir quel est l'ensemble minimal d'informations à stocker lorsque l'on réalise une expérience de puces à ADN, afin de pouvoir par la suite publier ses résultats, les échanger ou les comparer avec d'autres expériences du même type.
- **MAGE (MicroArray Gene Expression) :** il s'agit d'un standard pour la représentation des données de puces à ADN, c'est-à-dire un « Platform Independant Model » en terminologie MDA. Ce modèle est appelé MAGE-OM (MAGE Object Model) et a servi de base pour générer le langage MAGE-ML (MAGE Markup Language).

- **MGED Ontology :** il s'agit de la définition d'un vocabulaire standard, contrôlé, commun aux experts du domaine des puces à ADN. Une représentation des relations entre les termes de ce vocabulaire est également définie.

Il existe également au sein de la « MGED Society » des groupes de travail annexes qui concernent la normalisation des données (MGED Data Transformation and Normalization Working Group), l'extension des standards à d'autres type d'expériences de génomique fonctionnelle (Reporting Structure for Biological Investigations Working Groups), ainsi que l'extension de MIAME aux expériences de marquage et d'hybridation in situ (Minimum Information Specification For In Situ Hybridization and Immunohistochemistry Experiments).

Dans les paragraphes suivants, nous présentons plus en détail les trois axes du consortium MGED, en particulier le modèle MAGE-OM qui nous a servi de base pour notre travail de modélisation des données manipulées lors de la conception d'oligonucléotides.

4.2 MIAME (Minimum Information About a Microarray Experiment)

MIAME est un ensemble de recommandations qui spécifient les informations à stocker obligatoirement lorsque l'on effectue une expérience de puces à ADN [Brazma et al 2001]. L'objectif est de guider le développement des bases de données de puces à ADN, de faciliter l'interprétation et la reproductibilité des expériences, ainsi que l'échange et la comparaison des données.

MIAME se présente sous la forme d'une sorte de « check-list » divisée en deux grandes parties : description de la lame (Array design description) et description de l'expérience (Experiment description).

La partie description de la lame traite comme son nom l'indique de tout se qui concerne la fabrication du support solide constituant la puce à ADN : type de support, dimensions, type de sonde, informations sur les sondes, etc.

Voici la liste exhaustive de la partie description de la lame (il n'existe pas de traduction « normalisée » en français des termes MIAME) :

Array design description

1) Array related information

- array design name
- platform type: in situ synthesized, spotted or other
- surface and coating specification
- physical dimensions of array support (e.g. of slide)
- number of features on the array
- availability (e.g., for commercial arrays) or production protocol for custom made arrays

2a) For each reporter type

- the type of the reporter: synthetic oligo-nucleotides, PCR products, plasmids, colonies, other
- single or double stranded

2b) For each reporter

- sequence or PCR primer information:
- sequence or a reference sequence (e.g., for oligonucleotides), if known
- sequence accession number in DDBJ/EMBL/GenBank, if exists
- primer pair information, if relevant
- approximate lengths if exact sequence not known
- clone information, if relevant (clone ID, clone provider, date, availability)
- element generation protocol that includes sufficient information to reproduce the element for custom-made arrays that are not generally available

3a) For each feature type

- dimensions
- attachment (covalent/ionic/other)

3b) For each feature

- which reporter and the location on the array

4) For each composite sequence

- which reporters it contains
- the reference sequence
- gene name and links to appropriate databases (e.g., SWISS-PROT, or organism specific databases), if known and relevant

5) Control elements on the array

- position of the feature (the abstract coordinate on the array)
- control type (spiking, normalization, negative, positive)
- control qualifier (endogenous, exogenous)

La partie description de l'expérience concerne quant à elle les modalités de l'hybridation et du traitement des données : protocole expérimental de l'hybridation, préparation des cibles, acquisition des données, normalisation, etc.

Voici un extrait de la partie description de l'expérience qui concerne l'acquisition des données :

> ***Measurement data and specifications of data processing***
>
> *1) Raw data description should include*
>
> *- for each scan laboratory protocol for scanning, including scanning hardware and software, scan parameters, including laser power, spatial resolution, pixel space, PMT voltage;*
> *- scanned images;*
> *It should be noted that MGED does not have consensus whether the provision of images is a part of MIAME.*
>
> *2) Image analysis and quantitation*
>
> *- image analysis software specification and version, availability, and the description or identification of the algorithm and all the parameters used*
> *- for each image the complete image analysis output (of the particular image analysis software)*

Même s'il est possible de renseigner tous ces champs en langage naturel, MIAME recommande d'utiliser un vocabulaire contrôlé afin de faciliter les requêtes sur les bases de données et l'analyse automatique de texte (voir paragraphe 4.4).

La version complète de la spécification MIAME est disponible à l'adresse suivante :

http://www.mged.org/Workgroups/MIAME/miame_1.1.html

Comme tous les standards définis par le consortium MGED, MIAME est devenu une norme pour les expériences de puces à ADN, et il est indispensable de suivre ces recommandations. D'un point de vue logiciel, si l'on développe un programme ou une base de données propriétaire, il est nécessaire de prévoir le stockage de toutes les informations spécifiées. De même, si l'on utilise un logiciel existant, il est préférable d'en choisir un compatible MIAME. Le nombre de ces logiciels dits « MIAME compliant » ne cesse de croître, le Tableau 3 en présente quelques uns.

Nom du logiciel	Type	URL
Partisan arrayLIMS	LIMS[15]	http://www.clondiag.com/frame.php?page=/products/sw/partisan/
LIMaS [Webb et al 2004]	LIMS	http://www.mgu.har.mrc.ac.uk/facilities/microarray/limas/
ArrayHub	LIMS	http://www.integromics.com/products.htm
GenePix Pro 6.0	Analyse d'images	http://www.moleculardevices.com/pages/software/gn_genepix_pro.html
ArrayExpress et son outil de soumission MIAMExpress [Parkinson et al 2005]	Bases de données	www.ebi.ac.uk/arrayexpress/
BASE (BioArray Software Environment) [Saal et al 2002]	Bases de données	http://base.thep.lu.se/
Gene Traffic	Bases de données	http://www.stratagene.com/products/showProduct.aspx?pid=538
Genowiz	Analyse de données	http://ocimumbio.com/web/bioinformatics/
SAS microarray	Analyse de données	http://www.sas.com/industry/pharma/mas/

Tableau 3 : Les principaux logiciels de puces à ADN « MIAME compliant ».

4.3 MAGE-OM (MicroArray Gene Expression Object Model)

4.3.1 Historique

Lorsque l'on se situe dans le domaine de l'ingénierie des modèles et plus précisément dans le cadre d'une architecture logicielle de type MDA (Model Driven Architecture), MAGE-OM est un Platform Independent Model (PIM) du domaine des expériences de puces à ADN, décrit en UML [OMG 2003-a]. Son origine vient de la nécessité de définir un standard pour l'échange des données d'expériences entre la multitude de logiciels et bases de données existants. En effet, il ne suffit pas de définir quelles sont les informations à traiter (MIAME), mais il faut définir également la manière de les représenter.

Les premiers travaux ont commencé par définir des langages à balises dérivés de XML (eXtensible Markup Language) : MicroArray Mark-up Language (MAML) [EMBL 2000] et Gene-Expression Mark-up Language.

[15] Laboratory Information Management System : « cahier de laboratoire » électronique

Puis la spécification de MAGE-OM a été réalisée par le consortium MGED, en collaboration avec l'Object Management Group (OMG). En 1997, l'OMG crée en son sein le Life Science Research (LSR) group (www.omg.org/lsr/) qui a pour but de faciliter la communication et l'interopérabilité entre les ressources informatiques du domaine des sciences de la vie [Benton 2000]. Ce groupe de travail, composé d'acteurs de l'industrie pharmaceutique, d'universitaires, d'éditeurs de logiciels, opère dans le domaine des sciences de la vie au sens large : génomique, bioinformatique, chimie, biologie structurale entre autres. Aujourd'hui, son principal objectif est de définir des standards afin d'améliorer la qualité des logiciels et des systèmes d'information utilisés dans ces domaines, standards utilisant les technologies proposées ou soutenues par l'OMG : UML, CORBA, XML, EJB, et récemment l'approche MDA. Ces définitions de standards étaient au départ essentiellement composées de spécifications d'interfaces CORBA écrites en IDL (Interface Definition Language) et de modèles de domaines en XML. Avec l'arrivée de MDA, et la volonté de définir une norme indépendante d'un middleware particulier pour la création d'applications basées sur les composants, les spécifications du LSR group sont maintenant constituées de modèles de domaines indépendants de toute implémentation (PIM, Platform Independent Model) définis en UML.

Actuellement, dans le domaine de la bioinformatique au sens strict, le LSR group a adopté trois spécifications : Biomolecular Sequence Analysis (BSA) [OMG 2001], Genomic Maps [OMG 2002] et Gene Expression [OMG 2003-a]. Le standard BSA concerne le domaine central de la bioinformatique et de la génomique : l'analyse de séquences. Il se compose de deux sous modules : l'un modélisant les objets biologiques eux-mêmes (les séquences), et l'autre modélisant les mécanismes d'analyse de ces objets. Les interfaces de ces deux modules sont décrites en IDL. La spécification Genomic Maps adoptée en 2002 s'attache aux données de cartographie génétiques, et comprend, en plus des spécifications IDL, une ébauche de PIM. Enfin, dans le domaine des puces à ADN, la Gene Expression specification propose le PIM MAGE-OM pour la représentation et l'échange des données d'expression de gènes, ainsi qu'un PSM (Platform Specific Model) en XML.

4.3.2 Présentation générale

MAGE-OM est un modèle relativement complexe qui se veut suffisamment générique pour éventuellement englober des expériences d'autre type que l'hybridation. Il est composé de 132 classes groupées en 17 packages, contenant au total 123 attributs et 223 associations entre classes (Figure 18).

Les principaux packages sont :

- **BioSequence :** définit les classes décrivant tout ce qui concerne les séquences biologiques.
- **QuantitationType :** quantification du signal.
- **ArrayDesign :** plan de dépôt (container des classes de DesignElement).
- **DesignElement :** éléments du plan de dépôt.
- **Array :** décrit les procédures de fabrication de la lame à partir du plan de dépôt.
- **BioMaterial :** classes décrivant le matériel biologique servant à fabriquer les cibles.

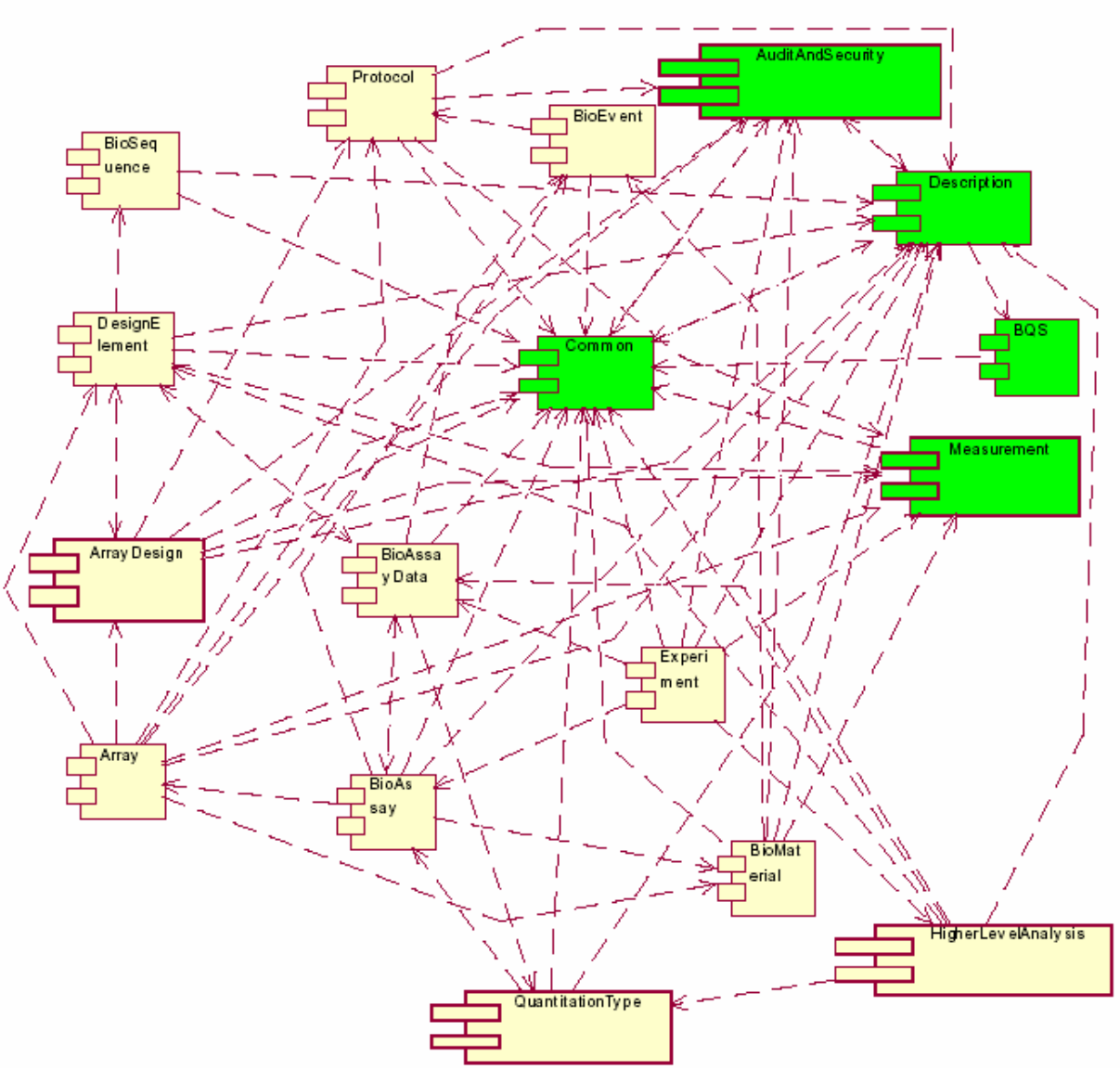


Figure 18 : Les packages du MAGE-OM et leur relations.

- **BioAssay :** classes décrivant le processus de mise en relation d'un Array (une lame) avec un BioMaterial (cible). Typiquement une hybridation dans le cas d'une expérience classique de puce à ADN.
- **BioAssayData :** données issues d'un BioAssay.
- **Experiment :** container pour les classes de BioAssay. Permet de grouper plusieurs hybridations dans le cas par exemple d'une étude temporelle.
- **HigherLevelAnalysis :** analyse des données BioAssaydata.
- **Protocol :** fournit des classes permettant de décrire de façon générale tout protocole expérimental.
- **Description :** classes servant à l'annotation de séquences ou de matériel biologique.
- **AuditAndSecurity :** spécifie tout ce qui concerne les droits et permissions sur les données.

- **Measurement :** définit des classes permettant de modéliser les mesures physiques (température, masse, volume…).
- **BioEvent :** englobe la classe abstraite *BioEvent* qui modélise un « traitement », qui possède des sources d'un certain type et produit des résultats d'un certain type (exemple : lavage de la lame, acquisition de l'image).

La Figure 18 montre la complexité des relations existant entre les différents packages. On voit que MAGE-OM est un modèle fortement couplé.

La Figure 19 représente la chaîne de traitement que constitue une expérience de puce à ADN. Elle permet de replacer les termes utilisés par le MAGE-OM dans leur contexte d'utilisation, notamment pour nommer les classes. La MGED Society vise à imposer un vocabulaire standard pour désigner tous les éléments d'une expérience de biopuce (voir paragraphe 4.4), et ces termes sont maintenant employés dans les principaux logiciels dédiés à ce type d'expérience.

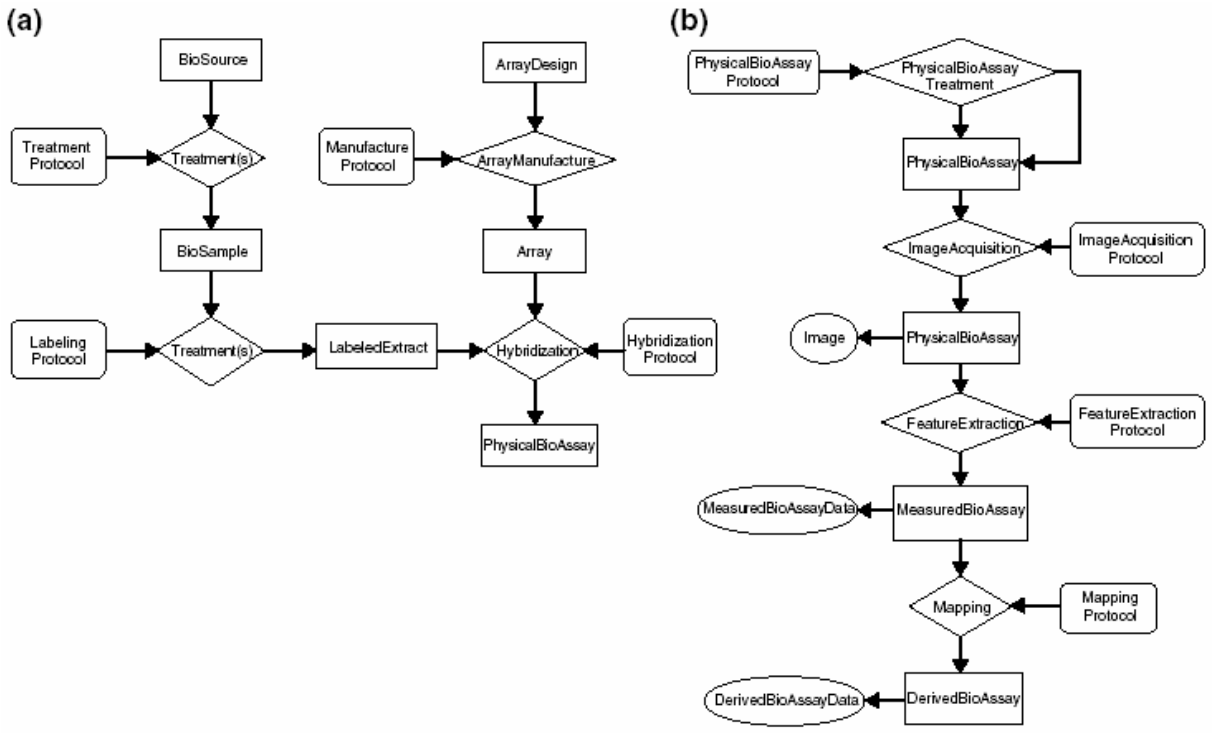


Figure 19 : Les différentes étapes d'une expérience de puce à ADN.

La partie (a) de la figure présente les étapes qui mènent à l'obtention d'une lame hybridée (**PhysicalBioAssay**). D'une part, il y a fabrication des cibles (**LabeledExtract**) par préparation et marquage d'échantillons biologiques (**BioSample**), et d'autre part, il y a fabrication de la lame (**Array**) à partir du plan de dépôt (**ArrayDesign**).

La partie (b) schématise le processus d'obtention des données. La lame hybridée peut subir un certain traitement (**PhysicalBioAssayTreatment**) comme le lavage. Ensuite, il y a acquisition des images (**ImageAcquisition**), puis extraction des données pour chaque spot

(**FeatureExtraction**), ce qui génère un objet de type **MeasuredBioAssay**. Il est également possible d'appliquer des transformations à ce dernier pour obtenir un objet contenant les données transformées (**DerivedBioAssay)**.

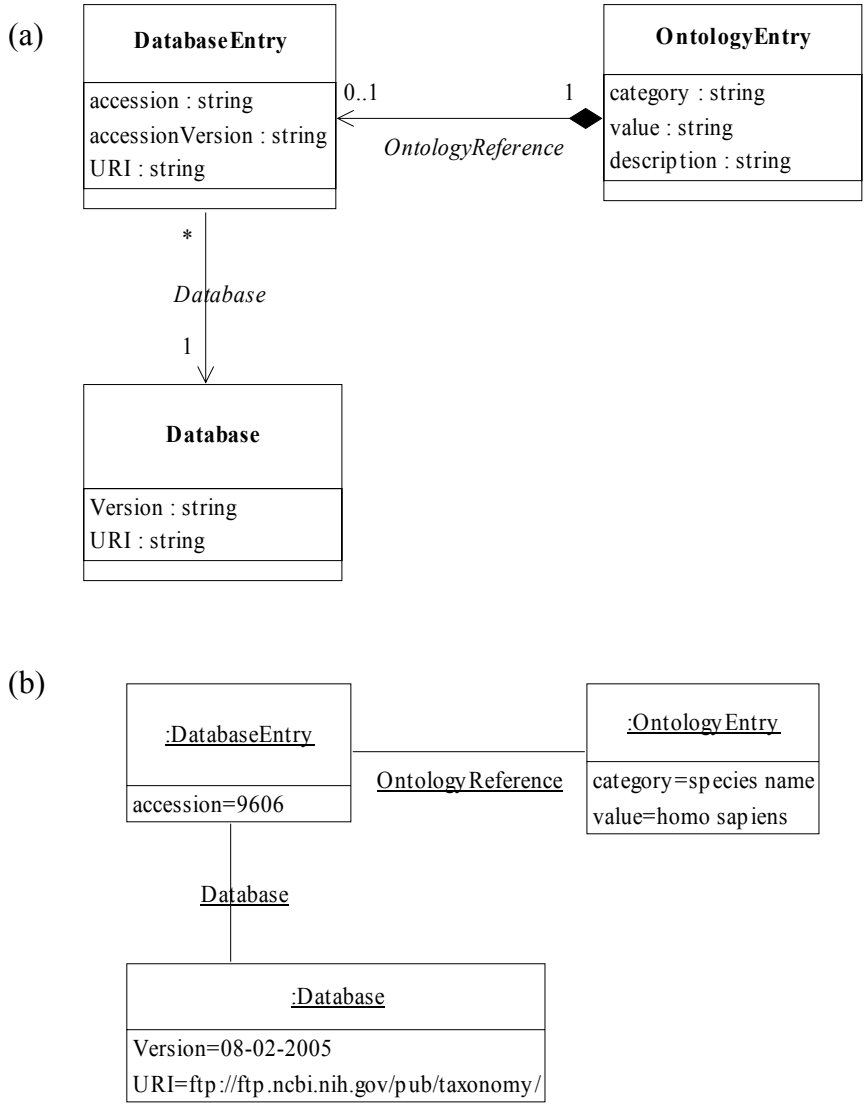


Figure 20 : La classe OntologyEntry du modèle MAGE-OM et les classes associées.

(a) Diagramme de classes
La classe OntologyEntry modélise une entrée dans une ontologie. Elle est associée à un enregistrement dans une base de données (DatabaseEntry), et cet enregistrement est associé à une base de donnée.
(b) Diagramme d'objets
Exemple d'instances dans le cas de la spécification d'un nom d'espèce.

En ce qui concerne l'utilisation d'un vocabulaire contrôlé, MAGE-OM se veut le plus générique et le plus évolutif possible, afin de pouvoir prendre en compte les modifications éventuelles des ontologies. Dans ce but, il définit une classe *OntologyEntry* (package Description) qui permet de modéliser une entrée dans une ontologie quelconque (Figure 20). Ainsi, supposons qu'une classe possède un attribut de type String par exemple, et que l'on

souhaite contrôler le vocabulaire utilisé pour renseigner cet attribut. Au niveau du modèle, l'attribut est remplacé par une association avec la classe *OntologyEntry*, et cette association prend le nom de l'attribut (Figure 21).

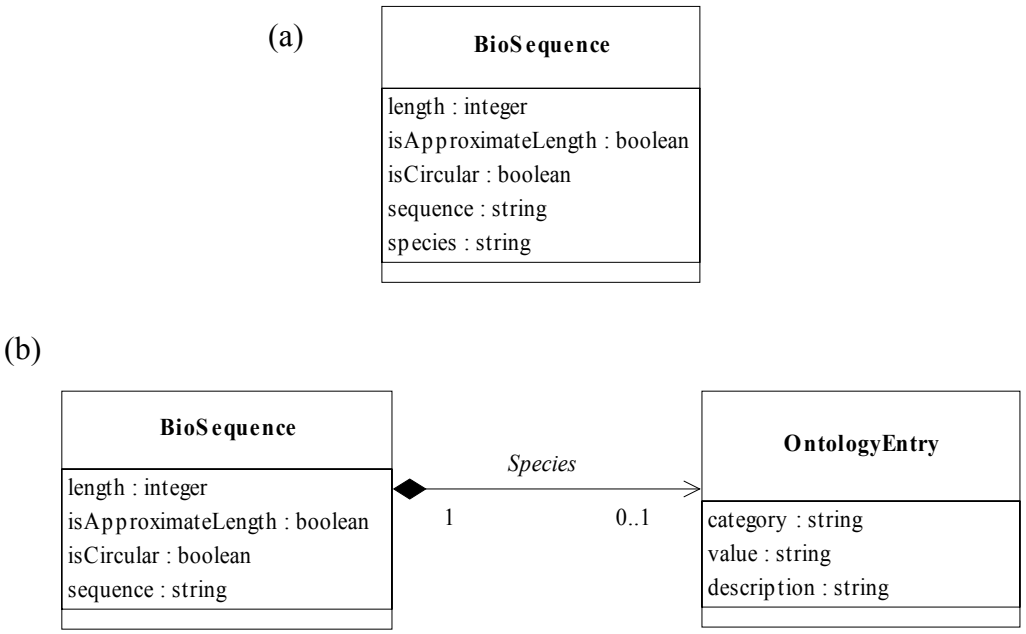


Figure 21 : Utilisation de la classe OntologyEntry pour introduire un contrôle du vocabulaire.

La classe BioSequence modélise une séquence biologique quelconque. Elle doit naturellement posséder une information sur l'espèce de l'organisme dont elle est issue. Plutôt que d'utiliser un attribut (a), le modèle MAGE-OM utilise une association (nommé *Species*) vers la classe OntologyEntry.

4.3.3 Position de la conception d'oligonucléotide dans le MAGE-OM

Dans le MAGE-OM, le terme utilisé pour désigner les sondes est **Reporter**. Ce terme tend également à se généraliser dans la communauté des puces à ADN, et remplace progressivement le mot « probe ». Les informations sur les sondes sont regroupées dans le package DesignElement. La Figure 22 présente l'ensemble des classes de ce package.

Les trois classes les plus importantes héritent de la classe abstraite *DesignElement*. Tout d'abord, la classe *Feature* désigne une localisation sur la lame en coordonnées absolues ou logiques (ligne, colonne), elle ne contient aucune information en terme de séquence biologique. La séquence biologique qui constitue la sonde est représentée par la classe *Reporter*. Ainsi, si une sonde est spottée plusieurs fois sur la puce en des endroits différents, un objet Reporter sera associé à plusieurs Feature (par l'intermédiaire de la classe *FeatureReporterMaps*). La classe *CompositeSequence* sert à modéliser le fait que pour étudier l'expression d'un gène, on peut avoir besoin de plusieurs sondes. Cette classe représentera alors le gène étudié et sera associée à plusieurs Reporter.

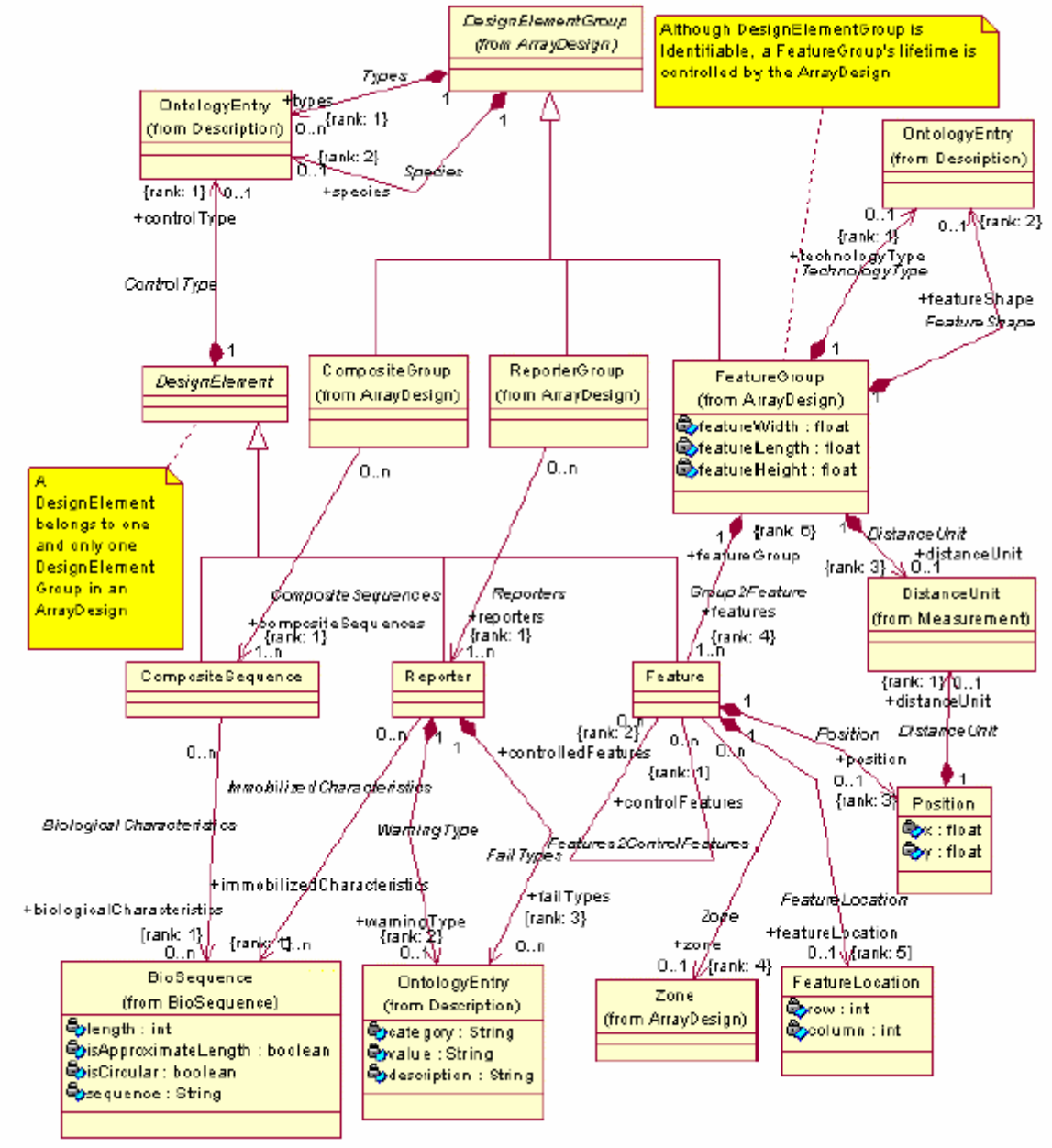


Figure 22 : Le package DesignElement.

La classe *Reporter* dispose des associations suivantes :

- **FailTypes : OntologyEntry (0..n)**
  Permet d'indiquer si la sonde est « défectueuse » et le type de problème.
- **WarningTypes : OntologyEntry (0..1)**
  Similaire à FailTypes sauf qu'il ne s'agit que d'un « warning ».
- **ImmobilizedCharacteristics : BioSequence (0..n)**
  La ou les séquences qui constituent cette sonde.
- **FeatureReporterMaps : FeatureReporterMap (0..n)**
  Associe une sonde à une ou plusieurs localisations sur la lame (Feature).

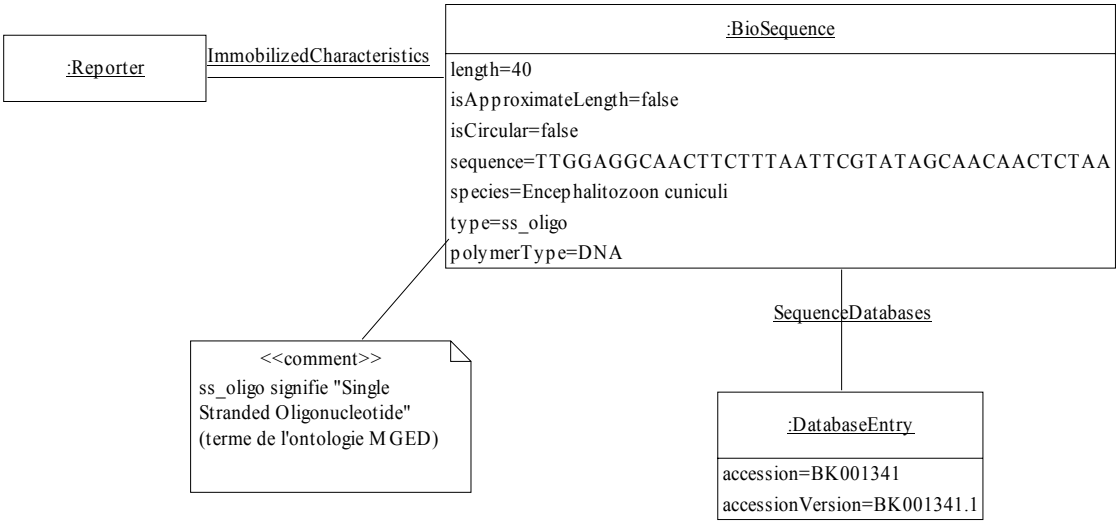


Figure 23 : Diagramme d'objets représentant une sonde oligonucléotidique de 40mers ciblant un gène d'encephalitozoon cuniculi.

(par soucis de simplification, les associations avec la classe OntologyEntry sont remplacées par des attributs de type string)

Le diagramme d'objets de la Figure 23 présente concrètement la manière dont est modélisée une sonde oligonucléotidique dans le MAGE-OM. La classe *Reporter*, modélisant la sonde, hérite de la classe *Identifiable* (non représentée sur la figure) comme beaucoup de classes du modèle. Cette classe *Identifiable* modélise toutes les entités possédant un nom et un identifiant. L'instance de *Reporter* est associée à une instance de *BioSequence* (ou plusieurs dans le rare cas où plusieurs séquences sont mélangées dans le même spot). C'est cette classe qui contient toutes les informations sur la séquence oligonucléotidique (longueur, suite de bases, liens avec une entrée dans une bases de données, …).

4.3.4 MAGE-ML

MAGE-ML est une DTD (Document Type Definition) XML, qui a été générée automatiquement à partir du MAGE-OM [Spellman et al 2002]. En terminologie MDA, il s'agit d'un PSM (Platform Specific Model). Son objectif est de définir un format standard de fichier (basé sur XML) permettant l'échange des données d'expériences de puces à ADN entre les différents logiciels et bases de données.

En effet, il existe à peu près autant de schémas conceptuels de bases de données de puces à ADN qu'il existe de bases de données de ce type (c'est-à-dire un très grand nombre !). Un laboratoire qui réalise des expériences de puce développe souvent une base de donnée locale avec son propre schéma bien adapté aux données traitées. Très vite apparaît le problème de l'importation de données provenant d'autres bases, en vue de comparer les résultats d'expériences par exemple. De même, la publication de résultats d'expériences de puces à ADN dans certaines revues telles que Nature implique au préalable la soumission des données à un entrepôt public [Ball et al 2004]. Pour le laboratoire survient alors le problème de l'exportation des données de sa base locale. Avec un format d'échange standard tel que MAGE-ML, il suffit alors lorsque l'on développe une base de données de prévoir les

fonctions d'importation et d'exportation dans ce format. L'interopérabilité entre les différents logiciels et entrepôts de données devient alors plus aisée (Figure 24).

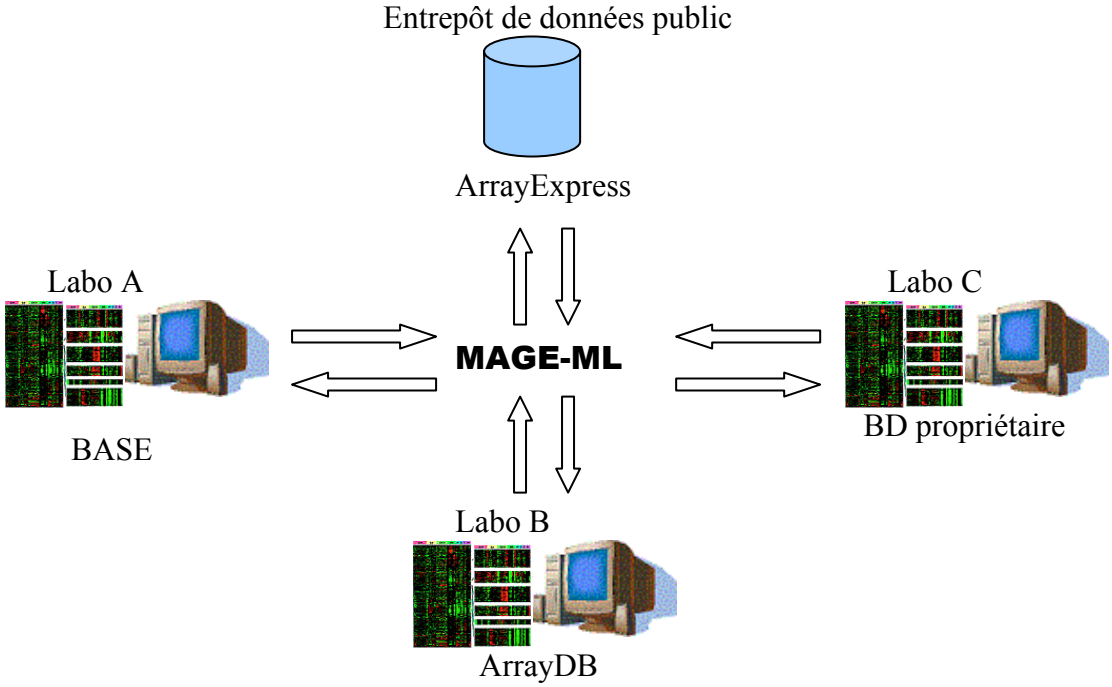


Figure 24 : Le format MAGE-ML permet l'échange de données d'expérience de puces à ADN.

Les deux principales règles qui ont été utilisées pour générer la DTD MAGE-ML sont les suivantes :

- Chaque classe du modèle objet est représentée par un élément avec une liste d'attributs identiques aux attributs de la classe.
- Pour chaque association d'une classe, un élément fils est créé. Cet élément prend le nom du rôle de la classe dans l'association suivi de « _assn ». Ensuite, si l'association est faîte par référence, « ref » est ajouté à la fin du nom, et si la cardinalité est supérieure à 1, « list » est ajouté à la fin du nom.

Pour plus de détails sur les règles de génération de la DTD, se référer à [OMG 2003-a].

Voici par exemple le fichier au format MAGE-ML correspondant au diagramme d'objet de la Figure 23 et donc représentant une sonde oligonucléotidique de 40mers ciblant un gène d'*Encephalitozoon cuniculi* :

```
<Reporter identifier="REP001341">
   <ImmobilizedCharacteristics_assnreflist>
      <BioSequence_ref identifier="BK001341"/>
   </ImmobilizedCharacteristics_assnreflist>
</Reporter>

<BioSequence identifier="BK001341" length="40"
isApproximateLength="false" isCircular="false"
sequence="TTGGAGGCAACTTCTTTAATTCGTATAGCAACAACTCTAA" >
   <SequenceDatabases_assnlist>
      <DatabaseEntry accession="BK001341"
                     accessionVersion="BK001341.1"/>
   </SequenceDatabases_assnlist>
   <PolymerType_assn>
      <OntologyEntry category="nucleic acid" value="DNA"/>
   </PolymerType_assn>
   <Type_assn>
      <OntologyEntry category="oligo" value="ss_oligo"/>
   </Type_assn>
   <Species_assn>
      <OntologyEntry category="species name"
                     value="Encephalitozoon cuniculi”/>
   </Species_assn>
</BioSequence>
```

4.3.5 MAGE-STK

MAGE-Software ToolKit est un ensemble de packages logiciels qui implémentent le modèle MAGE-OM dans divers langages [SpellMan et al 2002]. En mars 2005, il était disponible en open source pour les langages suivants : Java, Perl, Python et C#.

Il définit une API (Application Programming Interface) pour le MAGE-OM ainsi que des classes permettant de lire et d'écrire des fichiers au format MAGE-ML (Figure 25). Le MAGE-STK est disponible à l'adresse :

http://mged.sourceforge.net/software/MAGEstk.php

4.4 L'ontologie MGED

Comme nous l'avons vu dans les paragraphes précédents, MIAME et MAGE-OM ne suffisent pas à remplir les objectifs définis par la « MGED Society » en matière de standardisation de la représentation des données de puces à ADN. MIAME définit quelles sont les informations à stocker, MAGE-OM spécifie un modèle pour représenter ces informations, mais il est également nécessaire de définir les termes utilisés pour désigner les différents concepts. C'est l'objectif de l'ontologie MGED (MGED ontology).

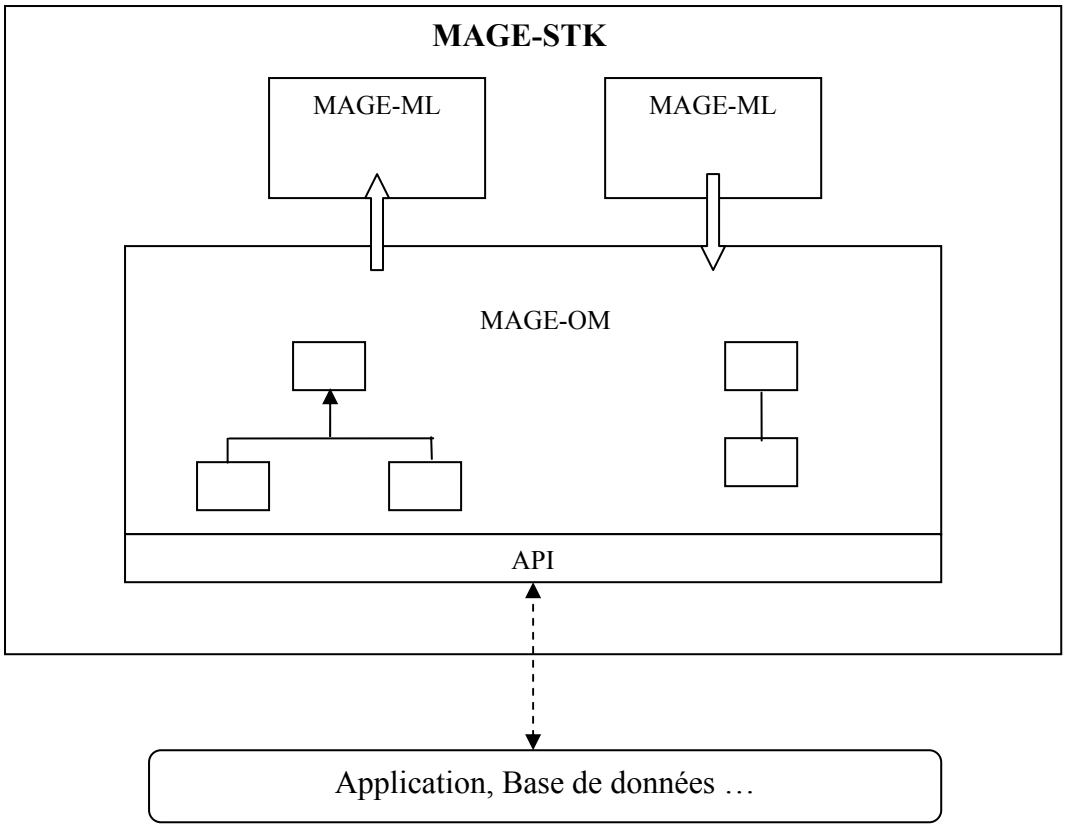


Figure 25 : Utilisation du MAGE-STK pour le développement d'une application.

L'API permet de « mapper » les données de l'application vers le MAGE-OM et ensuite de lire et d'écrire des fichiers au format MAGE-ML.

L'ontologie MGED est une ontologie du domaine des expériences de puces à ADN. Elle a pour but de fournir un vocabulaire contrôlé permettant de suivre les recommandations MIAME et de renseigner les attributs des classes du modèle MAGE [Stoeckert and Parkinson 2003].

Elle est basée sur les formalismes OIL (Ontology Inference Layer) [Fensel et al 2000] et DAML+OIL (DARPA[16] Agent Markup Language) [Horrocks et al 2002], qui sont deux langages spécialisés dans la définition d'ontologies. Elle est composée de trois types d'entités : des **classes**, des **propriétés** qui permettent de définir des relations de type composition entre les classes, et des **instances**. Il existe également des relations de type héritage entre les classes.

L'ontologie MGED est composée de 228 classes, 110 propriétés et 658 instances (version 1.2.0). Elle est disponible à l'adresse donnée ci-après, d'une part sous forme de page HTML, et d'autre part dans les formats utilisés par les éditeurs d'ontologies :

http://mged.sourceforge.net/ontologies/MGEDontology.php

[16] Defense Advanced Research Projects Agency : Département de recherche et développement de l'armée américaine.

Voici par exemple la description de la classe *Reporter* qui est le terme choisi par le consortium MGED pour désigner la séquence fixée sur le support solide (terme qui tend à remplacer « probe ») :

class **Reporter**

namespace:
http://mged.sourceforge.net/ontologies/MGEDOntology.daml#

documentation:
Description of the material placed on a feature (spot).

type:
  primitive

superclasses:
  DesignElement

constraints:
  restriction has_type has-class FailType
  restriction has_type has-class WarningType

De même, la classe *LabeledExtract* désigne les séquences marquées présentes dans le mélange d'hybridation (remplace l'ancien terme « target ») :

class **LabeledExtract**

namespace:
http://mged.sourceforge.net/ontologies/MGEDOntology.daml#

documentation:
The BioSample after labeling for detection of the nucleic acids.

type:
  primitive

superclasses:
  BioMaterial

constraints:
  restriction has_been_treated has-class Treatment
  restriction has_compound has-class LabelCompound

Un autre exemple permet d'illustrer les notions de propriété et d'instance. L'ontologie définit les classes *Image* et *ImageFormat* pour désigner respectivement une image de puce et un format d'image. La classe *Image* possède la propriété *has_image_format* qui indique qu'il existe une relation de composition entre la classe *Image* et la classe *ImageFormat* :

property **has_image_format**

namespace:
http://mged.sourceforge.net/ontologies/MGEDOntology.daml#

documentation:
property indicating that the class has an image format

domain:
PhysicalBioAssay

used in classes:
Image
PhysicalBioAssay

Un certain nombre d'instances de la classes *ImageFormat* sont définies : Affymetrix_DAT, GIF, JPEG, PNG, TIFF. Voici par exemple la définition de l'instance TIFF :

individual **TIFF**

namespace:
http://mged.sourceforge.net/ontologies/MGEDOntology.daml#

documentation:
Tag Image File Format (TIFF) is a common format to describe and store raster image data from scanners and other imaging devices. TIFFs may contain one or more channels and the data may be compressed using a lossless compression algorithm.

instance of:
ImageFormat

Tout comme pour MIAME, il est fortement recommandé lorsque l'on développe une base de données stockant des données d'expériences de puces à ADN, d'utiliser les termes de l'ontologie MGED. La plupart des logiciels « MIAME compliant » cités au paragraphe 4.2 se conforment à l'ontologie MGED.

5 Conclusion

Dans ce chapitre, nous avons vu que, dans les années à venir, le domaine du génie logiciel allait être fortement influencé par une nouvelle approche : l'Ingénierie Dirigée par les modèles, qui met les modèles, et non pas les programmes, au centre de la démarche de développement. Nous avons présenté plus particulièrement l'une des variantes de cette approche : MDA. Dans le domaine des puces à ADN, où il existait un manque de composants logiciels réutilisables, des efforts importants ont été entrepris en collaboration avec l'OMG pour appliquer les dernières techniques de Génie Logiciel. Ces efforts ont aboutis aux standards MGED avec notamment le modèle MAGE-OM.

Cependant, il est à noter que certaines caractéristiques inhérentes à la conception des oligonucléotides ne sont pas modélisées dans le MAGE-OM (température de fusion, hybridations croisées possibles,…). C'est ce constat, couplé à l'inefficacité des logiciels de conception de sondes existants, qui nous a amené à faire un certain nombre de propositions.

Dans le chapitre suivant, nous montrons tout d'abord pourquoi les algorithmes existant pour la conception de sondes sont peu adaptés à notre problématique biologique : l'étude de l'expression des gènes d'un parasite intracellulaire obligatoire (*Encephalitozoon cuniculi*). Nous proposons ensuite des approches permettant de résoudre les problèmes auxquels nous avons été confrontés. Nous présentons d'une part une nouvelle approche dans la conception d'oligonucléotide pour puces à ADN, et d'autre part un « Platform Independant Model » pour l'implémentation d'algorithmes de détermination de sondes.

Chapitre IV

Une nouvelle approche pour la conception de sondes
Proposition d'un « Platform Independant Model »

1 Introduction

Un des objectifs biologiques ayant guidé les travaux de cette thèse est la conception d'une biopuce destinée à suivre l'expression transcriptionnelle du pathogène *Encephalitozoon cuniculi*. Dans l'étape de recherche d'oligonucléotides spécifiques du modèle d'étude, notre première démarche a été de tester les logiciels libres existants. Nous avons alors été confrontés à deux problèmes principaux. D'une part, ces logiciels ne parviennent pas à déterminer d'oligonucléotide spécifique pour un très grand nombre de gènes d'*Encephalitozoon cuniculi*, notamment lorsque l'on spécifie que les cibles sont composées des transcrits du parasite mélangés à des transcrits humains. D'autre part, nous avons tenté d'adapter ces logiciels à notre problème en modifiant leur code source, disponible la plupart du temps. Cependant, ils sont souvent programmés avec une approche fonctionnelle et peu documentés, il est donc difficile d'effectuer une rétro ingénierie en vue de leur modification.

Dans ce chapitre, après avoir détaillé les différents problèmes évoqués ci-dessus, nous présentons des solutions sur le plan méthodologique avec une nouvelle approche pour la conception d'oligonucléotides. Sur le plan informatique, nous proposons un modèle objet qui s'inspire de la philosophie des modèles proposés par le MGED, et que nous avons étudiés au chapitre précédent.

2 Une nouvelle approche pour la conception de sondes

2.1 Les limites des logiciels existants

2.1.1 Le problème de la spécificité des sondes

Rappelons que l'objectif de l'étude biologique est de mieux comprendre les mécanismes d'adaptation impliqués dans le cycle de développement du parasite *Encephalitozoon cuniculi*. Le génome de ce pathogène est extrêmement compact (2,9 Mb) avec des régions intergéniques très courtes [Katinka et al 2001]. Il se compose d'environ 2000 gènes identifiés répartis sur 11 chromosomes. Pour les biologistes, l'idéal est de réaliser une puce à ADN permettant de mesurer l'expression de l'ensemble des gènes en une seule expérience, et donc de déterminer si possible un oligonucléotide spécifique pour chaque gène (le chapitre I présente l'étude biologique). La particularité d'une telle étude est que lors de la préparation des cibles, il est impossible de séparer les transcrits du parasite des transcrits de la cellule

hôte, en l'occurrence des transcrits humains dans notre cas. Cette particularité introduit une difficulté supplémentaire très importante dans la conception des sondes, car un olignucléotide ciblant un gène de *cuniculi* ne doit présenter aucune hybridation croisée avec un autre gène du parasite mais également aucune hybridation croisée avec un gène humain.

La plupart des logiciels étudiés utilisent le critère de Kane (voir chapitre II) pour tester la spécificité des oligos (OligoArray, OligoWiz, …). Même si ce critère n'a été définit que pour des sondes de 50mers, ils étendent ce résultat à des oligos de n'importe quelle longueur. Pour notre étude, nous avons choisi d'utiliser également le critère de Kane, car à notre connaissance, aucune autre étude biologique n'a été effectuée pour étudier les conditions d'hybridation croisée entre une sonde et un transcrit non-cible.

On rappelle donc qu'une sonde sera spécifique si et seulement si elle satisfait les deux conditions suivantes :

1) La séquence de l'oligonucléotide ne doit pas présenter plus de 75% de similarité (sur toute la longueur de la séquence) avec une séquence non-cible présente dans le mélange d'hybridation.

2) La séquence de l'oligonucléotide ne doit pas contenir une sous séquence de plus de 15 bases consécutives strictement identique à une séquence non-cible présente dans le mélange d'hybridation.

Un des seuls paramètres sur lequel on peut jouer pour tenter de trouver un plus grand nombre de sondes spécifiques est la longueur des oligonucléotides. On peut penser que plus la longueur est importante, plus la condition (1) sera facile à satisfaire. Mais il en va tout autrement pour ce qui est de la condition (2). En effet, plus la longueur augmente, plus la probabilité de trouver une séquence identique de 15 bases sera grande. Pour vérifier cette assertion et tenter de mieux comprendre la relation longueur des sondes-spécificité dans le cadre d'un modèle biologique complexe, nous avons effectué une étude *in silico* en utilisant l'algorithme BLAST.

2.1.2 Tests in silico de spécificité des sondes

Dans une première étude, nous recherchons des sondes spécifiques pour 120 gènes du chromosome 5 d'*Encephalitozoon cuniculi*. La recherche est effectuée en utilisant un algorithme très simple : étant donné la longueur recherchée *l*, on extrait tous les oligos potentiels de longueur *l* et on teste leur spécificité. On ne considère pas les autres critères afin de se concentrer sur la relation longueur-spécificité. Cette spécificité est déterminée en utilisant l'algorithme BLAST [Altschul et al 1997] avec les paramètres suivants :

- La taille du mot (W) est fixée à 7 (valeur minimale permise) afin que le maximum d'alignements soit détecté, même très courts.
- L'e-value (E) est fixée à 100 pour s'assurer également que tous les alignements intéressants seront dans le fichier résultat.
- Pour tenir compte du fait que le mélange cible est constitué des transcrits du parasite mélangés à des transcrits humain, la base de données BLAST est constituée de l'ensemble des CDS d'*Encephalitozoon cuniculi* ainsi que de la base de donnée UniGene [Wheeler et al 2004] pour l'humain.

Afin d'accélérer le calcul, on stoppe la recherche lorsque l'on a trouvé cinq sondes spécifiques. Les résultats sont présentés dans la Figure 26.

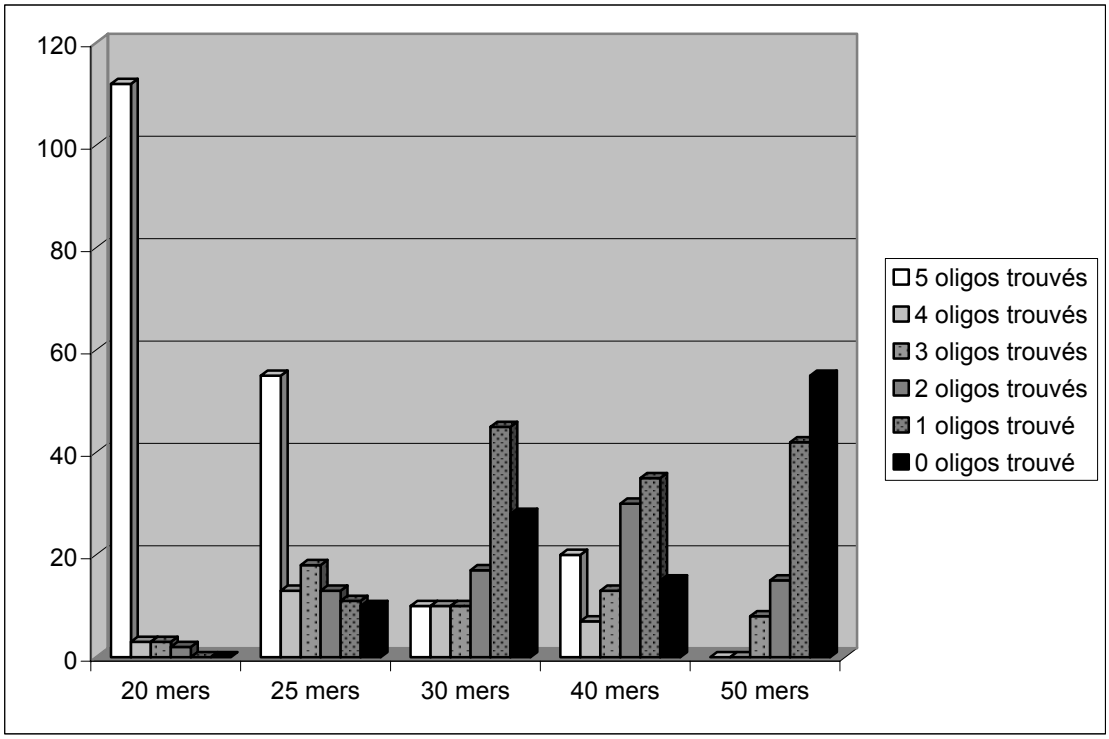


Figure 26 : Etude de la relation longueur des sondes - spécificité pour 120 gènes d'E. cuniculi

Pour 5 longueurs de sondes (20, 25, 30, 40, 50 mers), on recherche pour chaque gène au maximum 5 oligos spécifiques. Une colonne n de l'histogramme indique le nombre de gènes pour lesquels n oligos ont été trouvés.

Dans le cas où l'on recherche des sondes de 50 mers, on voit que pour un grand nombre de gènes, aucun oligo spécifique n'est trouvé. On trouve au maximum 3 oligos spécifiques pour quelques gènes. Pour achever la conception des sondes dans de telles conditions, il faudrait ensuite vérifier que ce petit nombre d'oligos spécifiques satisfait également tous les autres critères nécessaires pour avoir une sonde efficace, ce qui est peu probable. On voit également que si l'on diminue la longueur des sondes recherchées, la tendance s'inverse, on trouve de plus en plus d'oligos spécifiques. Ainsi, pour une longueur de 20 mers, on trouve 5 oligos spécifiques, c'est-à-dire le maximum recherché, pour 112 gènes sur 120. Il apparaît donc que dans notre contexte, plus la longueur des sondes est courte, plus il est possible de trouver des sondes qui vérifient le critère de Kane.

Afin d'élargir notre étude à l'ensemble du génome d'*E. cuniculi* ainsi qu'à d'autres organismes, nous avons mené une seconde série de tests selon le même principe. Nous utilisons cette fois comme base de donnée représentant le mélange cible d'une part le génome de la levure *Saccharomyces cerevisae*, et d'autre part notre modèle d'étude : l'ensemble des CDS d'*E. cuniculi* + la base de donnée UniGene pour l'humain. Pour chaque organisme, 100 CDS sont pris au hasard, et pour chacun de ces CDS, on mesure la spécificité d'un ensemble d'oligos potentiels (séquences disjointes). Pour ce faire, on utilise l'algorithme BLAST avec

la base de données représentant le mélange cible. On choisit comme mesure de spécificité la valeur suivante, déjà utilisée dans [Schroder et al 2001] :

$$Spécificité = \frac{Nbre\ hits}{Nbre\ oligos} * 100$$

Le nombre de « hits » est le nombre de gènes non-cibles pour lesquels l'oligo ne satisfait pas le critère de Kane. Pour un gène donné, le nombre d'oligos potentiels est différent suivant la longueur d'oligo choisie, c'est pourquoi on normalise le nombre de hits par rapport au nombre d'oligos testés.

La spécificité des oligos est mesurée en fonction de leur longueur. Les résultats sont représentés dans la Figure 27. La même tendance est observée pour les deux organismes : la spécificité des sondes de 20mers est très mauvaise, car ils présentent de très nombreuses hybridations croisées avec des gènes non-cibles. C'est pour une valeur située autour de 25-30 mers que la spécificité semble la meilleure, et à partir de cette valeur, plus la longueur augmente, plus la spécificité des sondes diminue. Cette tendance est d'ailleurs plus marquée dans le cas d'un mélange cible complexe (*E. cuniculi* + humain) que pour une étude classique (*Saccharomyces cerevisae*). Ceci confirme que dans le cas de sondes « longues », beaucoup d'hybridations croisées se produisent parce que ces sondes ne satisfont pas la condition (2) du critère de Kane.

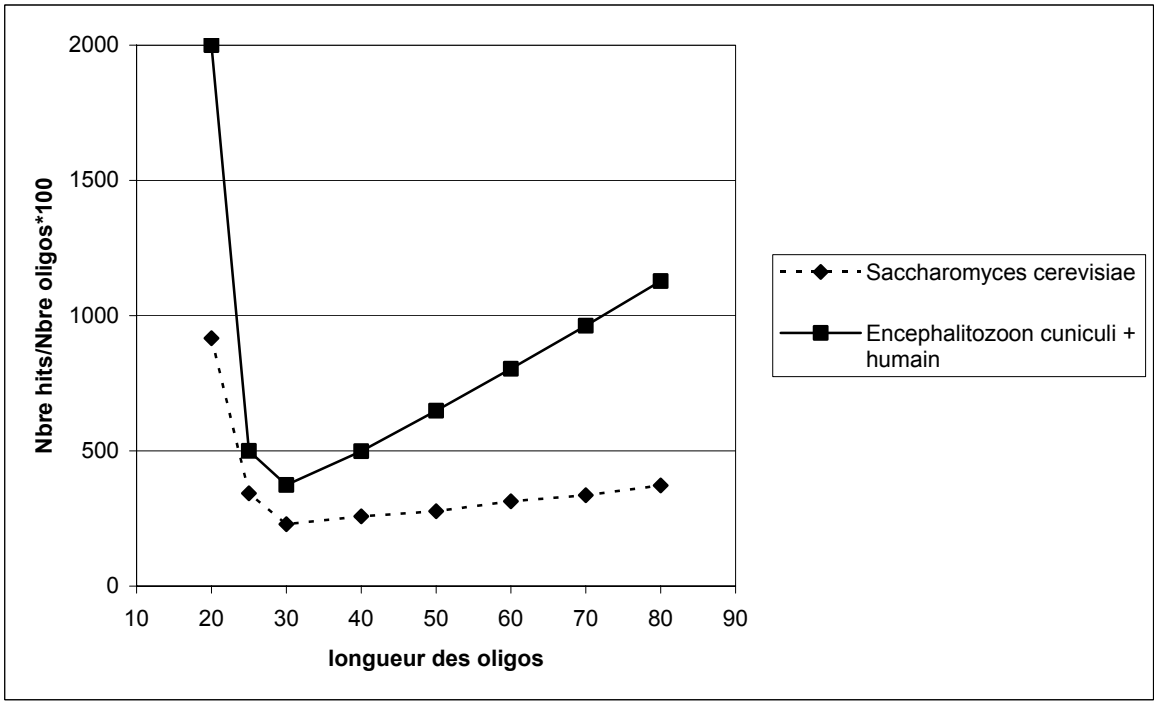


Figure 27 : Estimation de la relation longueur des oligos – spécificité dans le cas d'une étude classique (génome de *Saccharomyces cerevisae*) et d'un mélange cible complexe (*Encephalitozoon cuniculi* + humain).

2.2 Une nouvelle Approche

Les tests effectués ont mis en évidence le problème auquel nous étions confrontés : dans le cas d'un mélange cible complexe, correspondant aux génomes de plusieurs organismes, la longueur optimale des sondes se situe autour de 25-30 mers, afin d'assurer une bonne spécificité. Cependant, d'un point de vue biologique, les oligos de cette taille présentent une faible sensibilité de détection. La sensibilité est définie comme la quantité de signal émise par ARN présent dans le mélange cible [Binder et al 2004]. Elle représente la capacité à détecter de faibles quantités de transcrits. Le point clé dans la sélection d'oligonucléotides pour les puces à ADN est de choisir des sondes présentant une bonne spécificité grâce à des méthodes informatique (l'objet de cette thèse) tout en s'assurant qu'elles ont également une forte sensibilité (contraintes biologiques). Or il a été montré expérimentalement que les oligos court (20-30mers) présentent une faible sensibilité par rapport aux oligos de 50mers qui sont les plus couramment employés.

La nouvelle approche de détermination de sondes pour biopuces ADN que nous proposons ([Rimour et al 2005]) permet de respecter les deux contraintes : spécificité des sondes même dans le cas d'un mélange cible complexe, et sensibilité grâce à des sondes de longueur supérieure à 50mers. La séquence de la sonde est le résultat de la concaténation de 2 séquences « courtes » **disjointes** extraites du CDS (Figure 28). La détermination d'une sonde pour un gène donné revient alors à rechercher deux séquences spécifiques d'une longueur de 20-25 mers (plus facile à trouver qu'une séquence de 50mers d'après les tests ci-avant). L'hybridation avec la cible ADNc se fait alors avec formation d'une boucle dont la longueur dépend de la distance entre les deux sous-séquences. Dans la séquence de l'oligonucléotide, nous insérons également une séquence très courte (3-6 bases), que nous nommerons **linker**, entre les deux sous séquences, afin de faciliter la formation de la boucle. Ainsi, la séquence complète de l'oligonucléotide est relativement longue (par exemple 2 sous-séquences de 25mers combinées avec un linker de 5 bases soit 55 mers). Nous obtenons alors une sonde présentant une bonne sensibilité du fait de sa longueur, combinée à une excellente spécificité.

Nous avons implémenté cette approche originale de conception de sondes dans un logiciel appelé GoArrays. Ce logiciel sera présenté en détail au chapitre V. Dans la suite, nous nommerons « sondes GoArrays » les sondes déterminées par notre logiciel, conformément à la nouvelle approche.

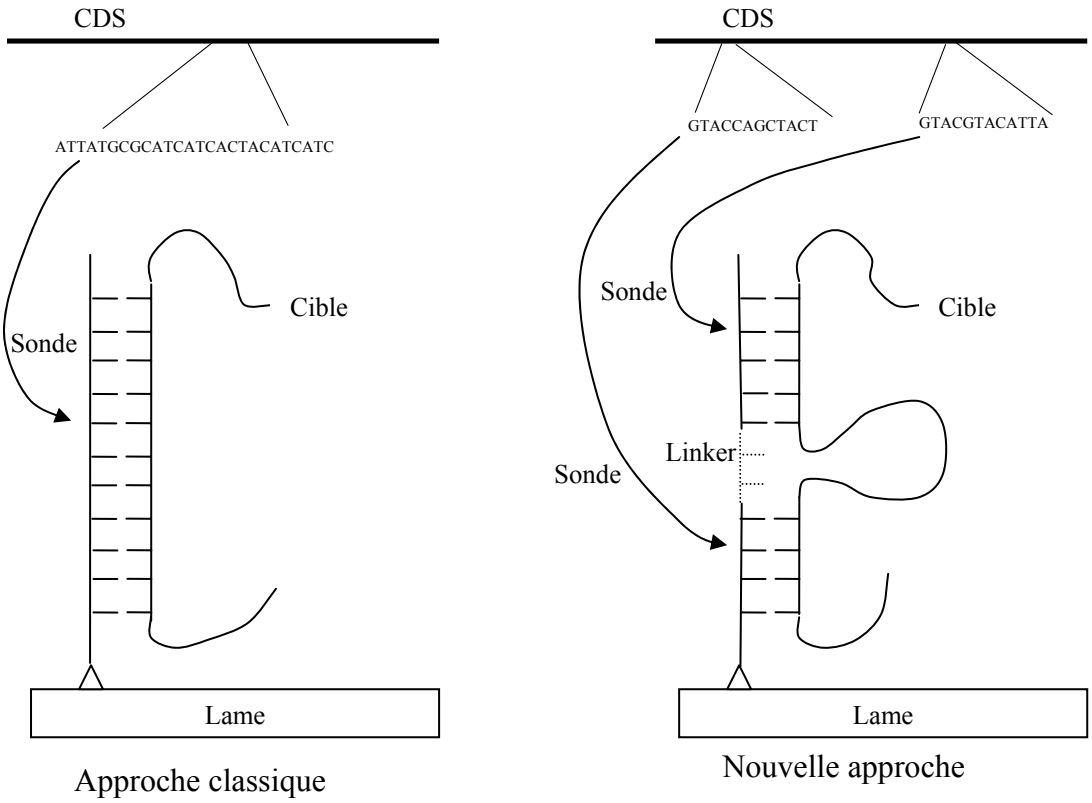


Figure 28 : Nouvelle approche pour la conception d'oligonucléotides pour biopuces ADN comparée à l'approche classique.

2.3 Vérification expérimentale

2.3.1 Présentation de l'étude

La validation de notre nouvelle approche pour la conception d'oligonucléotides nécessite bien évidemment une vérification expérimentale. En effet, l'utilisation de sondes s'hybridant avec une cible formant une boucle est complètement nouvelle. Il faut s'assurer que ces sondes présentent une sensibilité comparable aux sondes classiques tout en vérifiant que la spécificité est augmentée.

La première étude expérimentale que nous avons menée vise à vérifier ces propriétés dans des conditions d'hybridation très simples : nous nous concentrons uniquement sur deux gènes d'*Encephalitozoon cuniculi*. Les sondes déposées sur la puce ciblent uniquement ces deux gènes et le mélange cible est également composé seulement de transcrits provenant de ces deux gènes.

Les gènes étudiés sont d'une part un gène codant pour une protéine de tube polaire (PTP1) et d'autre part le gène de la cytidylate kinase (KCY). Le choix de PTP1 a été fait car il s'agit d'un gène fortement exprimé quelque soit les conditions expérimentales. Le choix de KCY

provient du fait que lorsque l'on réalise une recherche d'oligonucléotide spécifique pour PTP1 à l'aide d'un logiciel existant (OligoArray [Rouillard et al 2002]), le « meilleur » oligo trouvé présente tout de même un risque d'hybridation croisée avec le gène KCY. L'objectif de cette étude est donc de montrer que des sondes conçues à l'aide de notre nouvelle approche présentent une meilleure spécificité et qu'il est possible de déterminer une sonde pour PTP1 sans aucune hybridation croisée.

Nous avons donc effectué une recherche de sondes de longueur 50mers pour ces deux gènes à l'aide du logiciel OligoArray. Les résultats sont récapitulés dans le Tableau 4. La sonde trouvée pour PTP1 présente une homologie avec le gène KCY (16 bases identiques sur 17), il y a donc un risque d'hybridation croisée, selon les conditions d'hybridation. Il est important de noter que cette sonde retournée par le logiciel est celle qui présente le moins d'hybridations croisées potentielles, et donc que toutes les autres sondes recherchées avec une approche classique présenteront des hybridations croisées avec KCY. L'oligonucléotide déterminé pour KCY présente un risque d'hybridation croisée seulement avec un gène humain, et avec aucun gène d'*E. cuniculi*.

Gène	Séquence	Position dans le CDS	Hybridation croisée possible
PTP1	TAGGAACATGCAAGATTGCCGTATTGAAGCACTGCGACGCACCAGGAACA	896	KCY
KCY	AGGAACAACAGGGTATTCCTAGACGGAGAGGACGTGTCGGAGAGCCTCCG	450	gnl|UG|Hs#S1731224

Tableau 4: caractéristique des oligonucléotides de 50mers obtenus avec le logiciel OligoArray

Nous avons ensuite effectué une recherche d'oligonucléotides pour les gènes PTP1 et KCY à l'aide du logiciel GoArrays qui implémente notre nouvelle approche. 25 sondes par gènes ont été déterminées en faisant varier leurs caractéristiques : taille des deux sous-séquences constituant l'oligonucléotide chimérique, longueur du linker situé entre ces deux sous-séquences, seuil de spécificité considéré, longueur de la boucle formée par la cible lors de l'hybridation. Le seuil de spécificité permet de faire légèrement varier la condition (2) du critère de Kane. Soit S le seuil de spécificité exprimé en nombre de bases. La condition sera vérifiée si et seulement si :

> La séquence de l'oligonucléotide ne contient pas une sous séquence de plus de S bases consécutives strictement identique à une séquence non-cible présente dans le mélange d'hybridation.

En effet, [Rouillard et al 2002] proposent d'augmenter légèrement (16-17 bases) le seuil S par rapport à la condition (2) de Kane initiale, la condition (1) restant la même.

Les caractéristiques des 25 oligos calculés sont résumées dans le Tableau 5. Le paramètre « longueur des deux sous-séquences » n'est pas un paramètre que nous avons choisi mais il est déterminé directement par le logiciel : c'est la longueur pour laquelle la spécificité de la séquence est la meilleure. Pour des tailles de boucles de 10, 20 et 30 bases[17], nous avons fait

[17] Lorsque l'on indique une taille de boucle de 10 bases, le paramètre entré au logiciel est en fait 10-15 bases, pour faciliter la recherche et être moins strict. Donc au final, la taille de la boucle se situe entre 10 et 15 bases.

varier les deux autres paramètres de toutes les façons possibles avec 4, 5, ou 6 bases pour la longueur du linker et 15 ou 16 pour le seuil de spécificité. Nous avons également testé des tailles de boucles plus importantes en fixant les deux autres paramètres respectivement à 6 et 16. Pour identifier les oligos, nous les nommons de la façon suivante : PTP1_w_x_y_z pour le gène PTP1, avec w, x, y, z respectivement la taille de la boucle, la taille du linker, le seuil de spécificité et la longueur des deux sous-séquences. De même on nomme KCY_w_x_y_z_t les oligos ciblant le gène KCY. La Figure 29 détaille par exemple les caractéristiques de l'oligo PTP1_30_6_16_20. Notons enfin que chaque oligo déterminé par le logiciel GoArrays est spécifique au sens du critère de Kane.

Numéro oligo	Longueur boucle	Longueur linker	Seuil spécificité	Longueur sous-séquences
1	10	4	15	23
2	10	4	16	23
3	10	5	15	23
4	10	5	16	23
5	10	6	15	23
6	10	6	16	23
7	20	4	15	23
8	20	4	16	23
9	20	5	15	23
10	20	5	16	23
11	20	6	15	20
12	20	6	16	23
13	30	4	15	20
14	30	4	16	20
15	30	5	15	20
16	30	5	16	20
17	30	6	16	20
18	30	6	16	20
19	40	6	16	20
20	50	6	16	20
21	100	6	16	20
23	150	6	16	20
23	200	6	16	20
24	250	6	16	20
25	300	6	16	20

Tableau 5 : Caractéristiques des oligonucléotides chimériques obtenus avec le logiciel GoArrays.

Chacun des 25 oligos à été recherché à la fois pour le gène PTP1 et KCY.

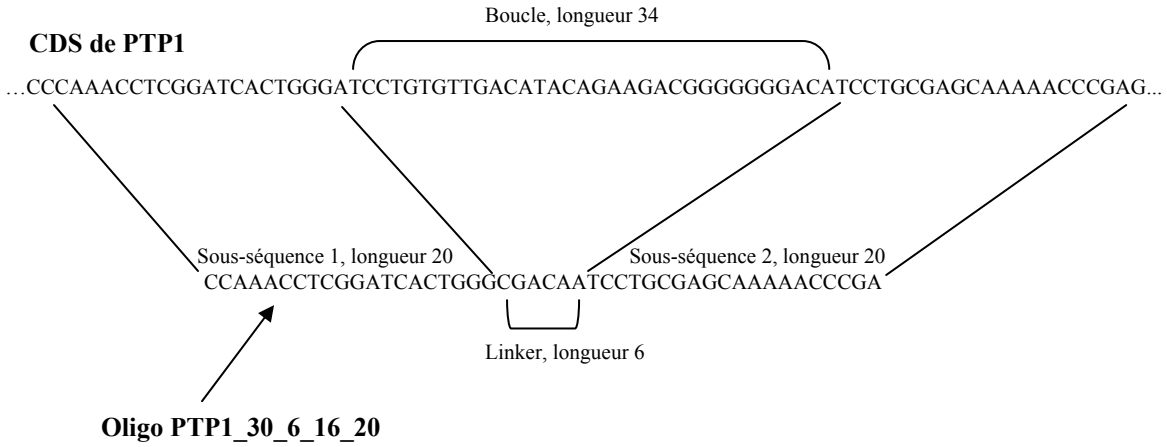


Figure 29 : Séquence et paramètres détaillés de l'oligo PTP1_30_6_16_20.

2.3.2 Protocole expérimental

Les sondes oligonucléotidiques ont été synthétisées par la société Eurogentec (http://www.eurogentec.be/). Les transcrits ARN des gènes PTP1 et KCY ont été obtenus par PCR (Polymerase Chain Reaction) suivie d'une transcription *in vitro*.

La transcription *in vitro* a été réalisée à l'aide du kit MEGAscript d'Ambion (http://www.ambion.com/). Puis le marquage des transcrtis ARN (3µg) avec le kit « Amino Allyl cDNA Labelling Kit » d'Ambion.

Les hybridations ont ensuite été réalisées en utilisant 25µl de mélange cible (17µl de solution tampon DigEasy de Boehringer, 5µl d'ADNc marqué, 3µl d'ADN de sperme de saumon à 1µg/ml) à 37°C pendant 16 heures dans une chambre d'hybridation Corning. Après hybridation, les lames ont été lavées deux fois à température ambiante pendant 5 minutes avec les solutions suivantes : 0.2X SSC, 0.1% SDS (solution 1), et 0.2X SSC (solution 2).

L'acquisition des images a été réalisée avec un scanner Affymetrix 428 (http://www.mwg-biotech.com/), et les données brutes ont été obtenues à l'aide du logiciel de traitement d'image Jaguar d'Affymetrix.

2.3.3 Résultats

Plusieurs hybridations ont été réalisées dans les conditions expérimentales décrites ci-avant. Sur une lame, chaque oligonucléotide est déposé en triplicat. Les valeurs considérées sont les moyennes des intensités de signal de chaque fluorochrome (Cy3/Cy5) sur l'ensemble des trois spots.

La Figure 30 présente un exemple d'images obtenues après hybridation. L'ARNm PTP1 était marqué avec le fluorochrome Cy3 et l'ARNm KCY avec Cy5. Ainsi « l'image Cy3 » permet de visualiser quelles sont les sondes qui ont hybridées avec les transcrits PTP1, et l'image Cy5 les sondes qui ont hybridé avec les transcrits KCY.

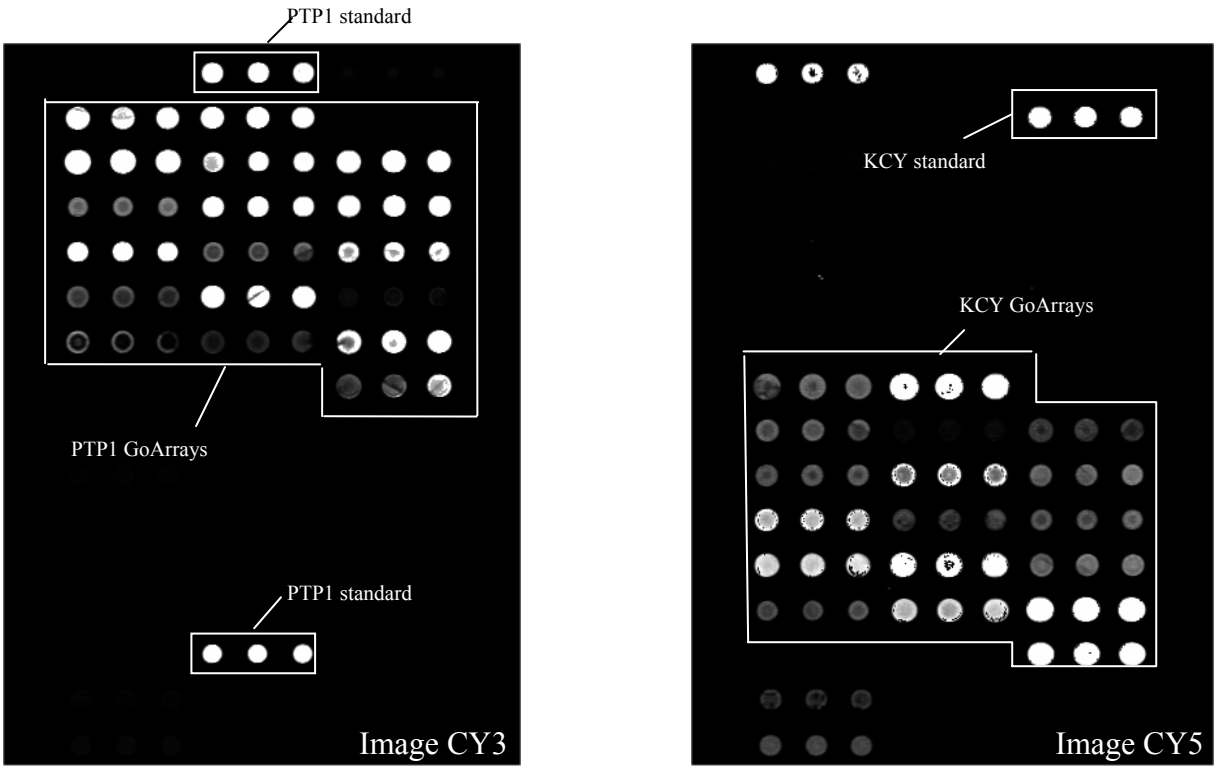


Figure 30 : Images obtenues après hybridation d'un mélange cible contenant des transcrits PTP1 marqués en Cy3 et des transcrits KCY marqués en Cy5 avec une lame test.

La lame test est composée de sonde classiques et de sondes déterminées avec notre nouvelle approche.

Les premières observations permettent de vérifier que les sondes obtenues à l'aide du logiciel OligoArray (sondes « standard ») hybrident bien avec leur cible. De plus, un certain nombre de sondes obtenues avec notre nouvelle approche (sonde « GoArrays ») hybrident également avec leur cible mais pas toutes. Ces observations sont confirmées par l'examen des données d'intensité obtenues avec le logiciel Jaguar (Tableau 6). Si l'on fait la moyenne des intensités de signal de toutes les sondes GoArrays, la valeur obtenue est très inférieure à celle des oligos standard et l'écart type est important. Cependant, certains oligos GoArrays présentent un signal comparable aux oligos standard, voire même supérieur (Tableau 7). Ceci nous permet de tirer une première conclusion : les oligonucléotides conçus avec notre nouvelle approche s'hybrident bien avec leur cible mais la sensibilité dépend de leur caractéristiques (longueur de la boucle formée lors de l'hybridation, longueur du linker…).

	Intensité du signal pour les oligos standard		Intensité du signal pour l'ensemble des oligos GoArrays	
	Moyenne	Ecart type	Moyenne	Ecart type
PTP1	18277	1213	14989	9793
KCY	44231	3744	13793	9638

Tableau 6 : Comparaison des intensités de signal obtenues pour les oligos standard et les oligos GoArrays.

Identifiant oligo	Intensité du signal	
	Moyenne	Ecart type
KCY_BACSU_30_3_16_20	42011	2244
KCY_BACSU_10_5_16_20	33896	1988
PTP1_30_4_16_20	31611	1854
PTP1_30_5_16_20	18855	1140

Tableau 7 : Intensité de signal obtenue avec quelques sondes GoArrays.

La sensibiltié est dans ce cas comparable, voire même meilleure qu'avec des sondes classiques

Il est donc nécessaire de comprendre comment les caractéristiques de nos sondes GoArrays influent sur la sensibilité de l'hybridation avec les cibles.

Etude de l'influence des caractéristiques des sondes GoArrays sur la sensibilité de l'hybridation

Le Tableau 8 présente les caractéristiques de 6 oligos GoArrays : trois d'entres eux possèdent une sensibilité élevée (intensité de signal supérieure aux oligos standards), alors que les trois autres affichent un signal très faible. On note que ces derniers possèdent soit un faible pourcentage en GC (pour au moins une des deux sous-séquences), soit un linker très court (une base). De plus, d'autres analyses des données brutes montrent qu'aucun oligonucléotide ne s'hybride avec sa cible si une boucle de plus de 100 bases doit se former (déstabilisation de l'hybridation). Nous concluons donc qu'une sonde déterminée avec notre nouvelle approche doit posséder une longueur de boucle inférieure à 100 bases, une taille de linker supérieure à 3 bases et un pourcentage en GC suffisant.

Vérification de l'hybridation de la cible sur toute la longueur de la sonde

Etant donné la configuration particulière de l'hybridation sonde-cible avec notre nouvelle approche, il est essentiel de vérifier que la séquence cible s'hybride bien avec les deux sous-séquences de la sonde et pas seulement avec l'une d'entre elles. En effet, dans ce cas, il est inutile d'avoir une sonde formée de deux séquences disjointes si l'hybridation ne se fait qu'avec une seule. Cette situation est schématisée Figure 31.

Identifiant oligo	Intensité du signal	%GC 1ère sous -séquence	%GC 2ème sous-séquence	Taille linker
KCY_BACSU_30_3_16_20	42011	60	45	3
PTP1_30_4_16_20	31611	60	55	4
PTP1_40_6_16_20	31403	60	40	6
KCY_BACSU_10_4_16_20	6222	35	45	4
PTP1_20_6_15_20	6670	70	35	6
PTP1_20_1_15_23	662	65	45	1

Tableau 8 : Comparaison des caractéristiques des oligonucléotides GoArrays.

L'identifiant de la sonde indique la taille de la boucle. Les colonnes suivantes indiquent l'intensité de signal, le pourcentage en GC des deux sous-séquences ainsi que la taille du linker.

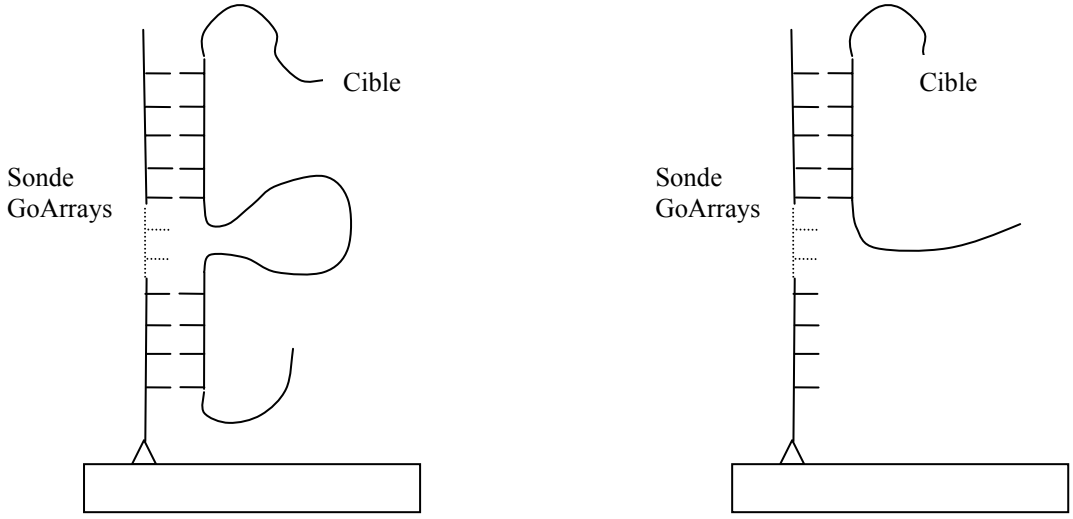


Figure 31 : Schéma d'une hybridation normale entre la séquence cible et la sonde GoArrays, et d'une hybridation partielle à éviter.

Même ci cette situation est peu probable du fait de la différence de température de fusion entre les deux configurations, il est nécessaire de vérifier qu'elle ne se produit pas. Pour ce faire, d'autres types de sondes ont été déposés sur la lame test. Etant donné une sonde GoArrays, nous avons déposé 3 sondes supplémentaires : chacune des deux sous-séquences de 20 bases composant l'oligonucléotide chimérique, ainsi qu'une sonde de même longueur que la sonde GoArrays, mais composée d'une séquence aléatoire accolée à l'une des deux sous-séquences. Ceci est récapitulé Figure 32. Nous avons répété ces tests pour 8 sondes GoArrays et les résultats sont présentés dans le Tableau 9.

Oligonucleotide GoArrays	Intensité du signal	Intensité du signal de la sonde test 1 correspondante (20mers, séquence A)	Intensité du signal de la sonde test 2 correspondante (20mers, séquence B)	Intensité du signal de la sonde test 3 (48mers, séquence aléatoire + séquence A)
KCY_BACSU_10_4_15_20	32842	1355	20805	5176
KCY_BACSU_10_5_16_20	33896	-	-	-
KCY_BACSU_30_3_16_20	42011	10724	1820	8052
KCY_BACSU_100_6_16_20	33919	-		-
PTP1_30_4_16_20	31611	412	6291	2146
PTP1_30_5_16_20	31683	-	-	-
PTP1_30_6_16_20	33499	2575	11	4051
PTP1_40_6_16_20	31403	-	-	-

Tableau 9 : Comparaison des intensités de signal des sondes GoArrays et des sondes test correspondantes.

Dans le cas où aucune valeur n'est indiquée, le signal n'a pas été mesuré (trop faible, ou segmentation du spot impossible)

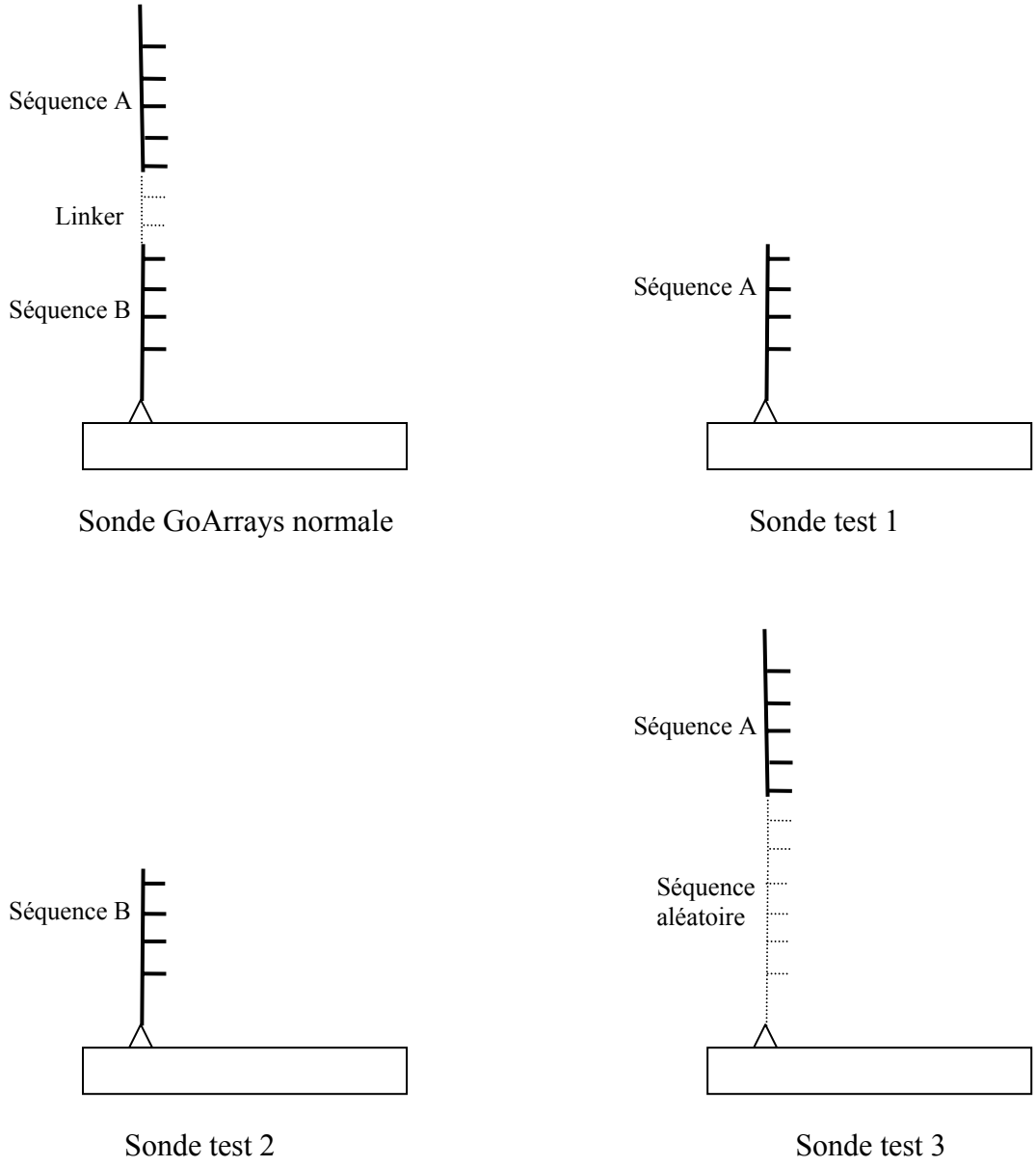


Figure 32 : Sondes test déposées afin de vérifier que l'hybridation de la séquence cible a bien lieu sur toute la longueur de l'oligo.

Pour la sonde test 3, la séquence A est placée soit du côté de la lame, soit à l'extrémité de la sonde comme sur le schéma.

On voit que dans presque tous les cas, le signal des sondes test reste très faible, voire nul. Pour seulement deux oligonucléotides, on mesure un signal significatif mais qui reste inférieur au signal de la sonde GoArrays complète (KCY_BACSU_10_4_15_20 et KCY_BACSU_30_3_16_20 avec respectivement 63% et 25% de la sonde complète). Ces résultats confirment le fait que l'hybridation de la séquence cible se produit bien avec l'ensemble de l'oligonucléotide chimérique avec formation d'une boucle comme nous l'avions prévu.

Etude de la spécificité des sondes GoArrays par rapport aux sondes classiques

Il est également nécessaire d'étudier la spécificité des oligonucléotides conçus avec notre nouvelle approche. En effet, la principale caractéristique de ces sondes doit être d'apporter une meilleure spécificité par rapport aux sondes classiques, dans le cas d'un mélange cible complexe.

L'hybridation de la lame test décrite dans les paragraphes précédents a été répétée plusieurs fois, en mesurant le signal Cy5 obtenu avec les sondes PTP1. La mesure d'un signal significatif avec ces sondes met en lumière la présence d'une hybridation croisée avec les transcrits non-cible KCY. Lors de certaines expériences, un signal non négligeable a été mesuré (Tableau 10) pour les sondes PTP1 standard, ce qui confirme les résultats des tests *in silico*, c'est-à-dire le risque d'hybridation croisée de l'oligonucléotide PTP1 conçu avec l'approche standard. Pour les sondes GoArrays, aucun signal d'hybridation croisée n'a jamais été détecté.

	Intensité du signal	Bruit de fond
PTP1 standard	1361	93
PTP1_30_6_16_20	no signal	53
PTP1_30_5_16_20	no signal	50

Tableau 10 : Mesure de l'intensité du signal Cy5 (cibles KCY) pour les sondes PTP1.

La présence d'un signal non néligeable indique une hybridation croisée des sondes conçues avec l'approche standard.

Ces résultats permettent de valider notre postulat de départ : les oligonucléotides chimériques déterminés grâce à notre nouvelle approche possèdent une spécificité accrue par rapport aux sondes classiques. Il faut tout de même modérer cette affirmation en notant que la vérification expérimentale a été réalisée avec un mélange cible réel très simple. Cependant, le mélange cible théorique qui a servi aux tests d'hybridations croisées était un mélange complexe (*Encephalitozoon cuniculi* + *humain*).

2.4 Bilan

Dans cette première partie du chapitre concernant nos propositions, nous avons répondu au problème de spécificité des sondes auquel nous étions confronté. Nous avons proposé une approche originale tant sur le plan biologique que méthodologique pour la détermination des oligonucléotides. Cependant, rappelons que nous souhaitions adopter une approche interdisciplinaire, et donc apporter une contribution sur le plan informatique pur, et plus particulièrement dans le domaine du Génie Logiciel. Cette contribution fait l'objet du paragraphe suivant.

3 Un « Platform Independant Model » pour le design d'oligonucleotides

3.1 Les problèmes rencontrés avec les logiciels existants

Nous avons vu dans les chapitres précédents que beaucoup de logiciels de design d'oligonucléotides pour puces à ADN existants étaient difficilement modifiables directement pour les adapter à notre problématique. En effet, ils sont souvent programmés selon une approche de programmation fonctionnelle, même si un langage à objets a été utilisé. De plus, la documentation se réduit souvent à sa plus simple expression. Il nous est donc apparu difficile et peu intéressant sur le plan de la réutilisabilité d'implémenter notre approche originale dans un logiciel existant. Nous avons vu également que des efforts avaient été entrepris par la communauté bioinformatique pour utiliser les dernières techniques de génie logiciel soutenues par l'OMG dans le domaine des puces à ADN.

Nous avons donc choisi de concevoir une nouvelle application de conception d'oligonucléotides (implémentant notre approche originale) en suivant la « philosophie » proposée par l'OMG dans le cadre du MDA. Dans les paragraphes suivants, nous proposons un « Platform Independant Model » pour la conception d'oligonucléotides. Ce modèle sera constitué de diagrammes de classes UML représentant les entités appartenant au domaine considéré, indépendamment de toute plate-forme d'exécution. Il utilise également certains packages existant dans le PIM MAGE-OM, et il est donc de ce fait totalement intégré avec celui-ci.

3.2 Rétro ingénierie du logiciel OligoArray

La première étape de notre travail a été d'effectuer une rétro ingénierie du logiciel OligoArray [Rouillard et al 2002]. Ce logiciel est en effet assez représentatif des logiciels de conception d'oligonucléotides : son étude nous a aidé à recenser l'ensemble des objets manipulés lors d'un processus de design de sondes et à concevoir notre PIM. D'un point de vue algorithmique, il emploie une méthode assez classique de parcours des CDS à partir de l'extrémité 5', en testant si les sondes potentielles vérifient les différents critères décrits au chapitre II. Il est assez simple d'utilisation et son code source est disponible.

Bien que programmé en Java, OligoArray n'est pas implémenté selon une approche orientée objet. Il est constitué d'une seule classe, et utilise une approche fonctionnelle. L'étude effectuée nous a permis de proposer un diagramme de classes UML des entités manipulées lors de son processus de design de sondes (Figure 33) [Hill et al 2002].

Les deux principales classes sont **Cds** et **Oligo**, qui héritent toutes les deux de la classe **BioSequence**. Cette dernière est une classe du MAGE-OM. Elle permet de représenter une séquence biologique quelconque. Le but du design d'oligonucléotide est de trouver, pour un CDS, une ou plusieurs séquences (oligonucléotides) spécifiques de ce Cds. Un CDS donné est composé de 1 à n oligonucléotides, car il peut être nécessaire de déterminer plusieurs oligos pour un même CDS. On peut manipuler soit des oligonucléotides spécifiques (**SpecificOligo**)

soit des oligonucléotides non spécifiques (**NonSpecificOligo**) qui héritent de la classe **Oligo**. Les possibilités d'hybridations croisées entre une sonde et un CDS non-cible sont représentées par l'association entre les classes Cds et Oligo. Le logiciel OligoArray introduit également deux types d'hybridations croisées : « sister » cross-hybridation et « cousin » cross-hybridation. Un oligo susceptible d'hybrider avec un CDS non cible est « sister » ou « cousin » de ce CDS suivant le degré de similarité entre les séquences.

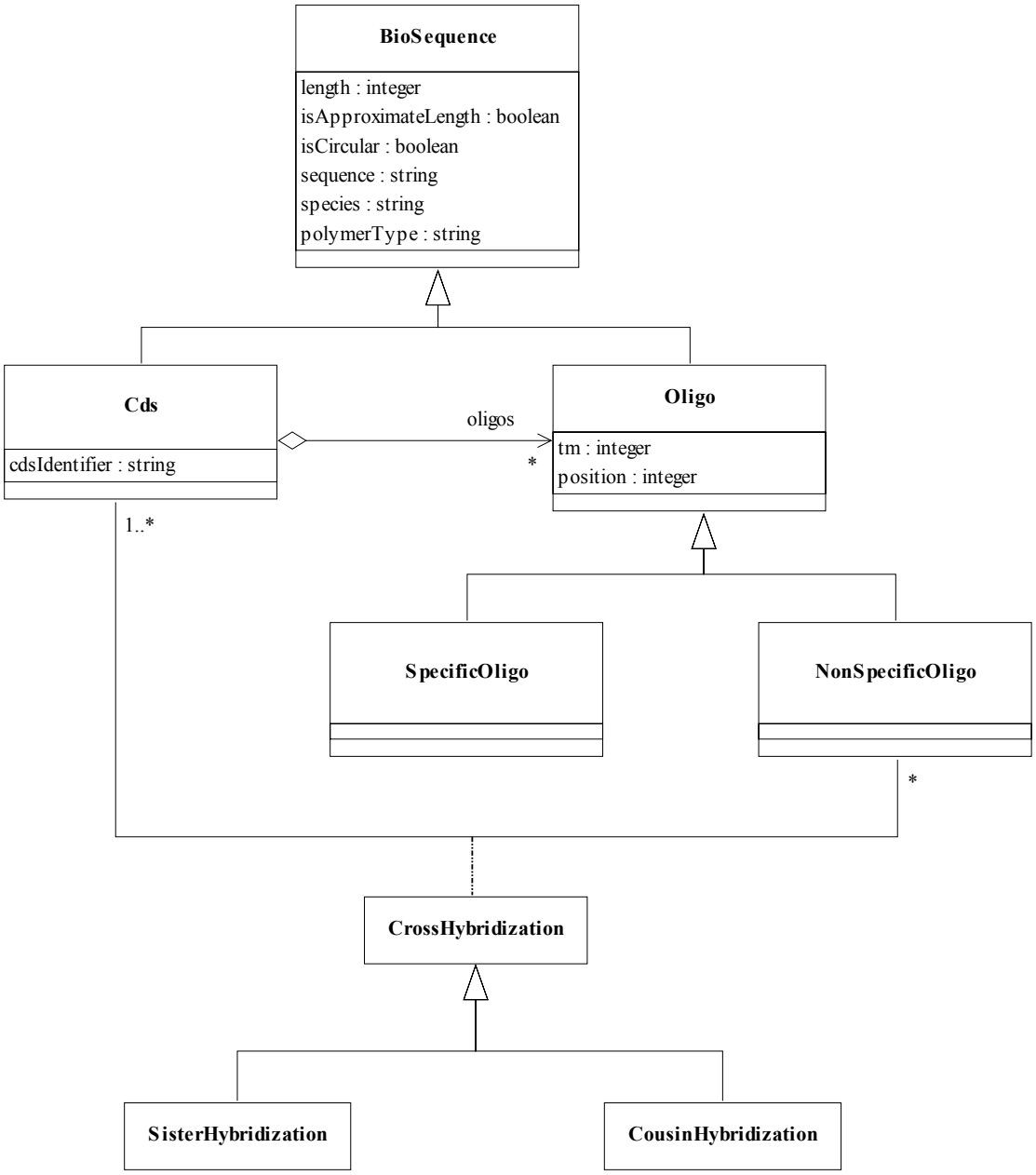


Figure 33 : Diagramme de classes issu du reverse engineering du logiciel OligoArray.

3.3 Proposition d'un PIM pour la conception d'oligonucléotides

L'étude du logiciel OligoArray nous a permis d'ébaucher un premier modèle du domaine de la conception d'oligonucléotide pour puces à ADN. Nous avons ensuite affiné ce modèle en introduisant les classes nécessaires à l'utilisation de notre nouvelle approche : l'utilisation de sondes chimériques. Nous souhaitions également étendre le modèle MAGE-OM, notamment en réutilisant certaines de ces classes.

Nous proposons donc un Platform Independant Model constitué de deux packages représentant le domaine des conceptions de sondes pour puces à ADN. Le package **Oligonucleotide** modélise les différentes entités manipulées lors du processus. Le package **DesignMethod** quant à lui modélise les algorithmes mis en œuvre.

3.3.1 Intégration avec MAGE-OM

Les deux packages utilisent des classes existantes dans le modèle MAGE-OM, notamment la classe BioSequence du package BioSequence, utilisée pour représenter une séquence biologique et la classe Database du package Description, utilisée pour représenter une base de données de séquences (Figure 34).

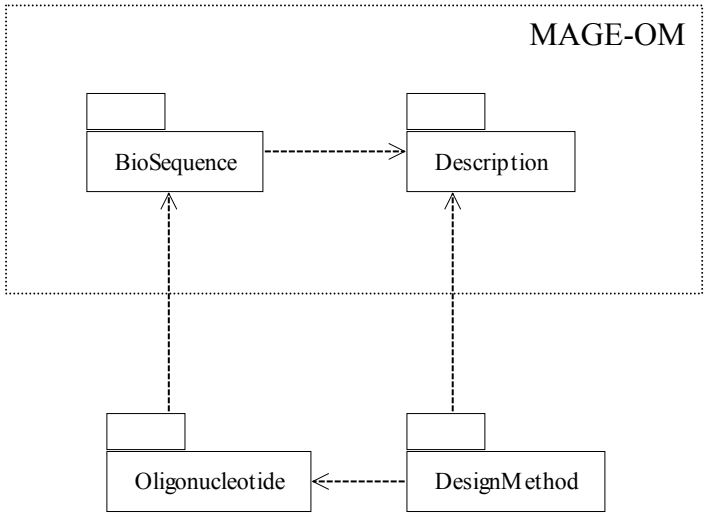


Figure 34 : Position du modèle proposé dans le MAGE-OM.

3.3.2 Le package Oligonucleotide

Le package Oligonucleotide fournit des classes permettant de manipuler les différents types de séquences biologiques utilisées lors d'un processus de design de sondes (Figure 35). Les sondes manipulées peuvent être de type classiques ou chimériques, conformément à notre nouvelle approche.

Par rapport au diagramme de classes issu de la rétro ingénierie du logiciel OligoArray, plusieurs modifications ont été apportées. Premièrement, un oligonucléotide « classique » (non chimérique) est représenté par la classe **Reporter** du MAGE-OM. En effet,

contrairement à OligoArray qui considérait une sonde comme une sous-classe de BioSequence, nous utilisons dans notre PIM la classe proposée par le MAGE-OM pour représenter une sonde. Un **DesignGroup** est un ensemble de gène ou Cds pour lesquels on souhaite déterminer des sondes. Il est donc composé de un ou plusieurs **Cds**. Un Cds est lui-même composé d'une ou plusieurs sondes (Reporter), représentatives de ce Cds.

La distinction entre oligo spécifique et non spécifique est supprimée au niveau des classes par rapport au modèle « OligoArray ». Elle est représentée par l'association **CrossHybridization** qui symbolise une hybridation potentielle entre une sonde et un Cds non-cible. On a également supprimé la distinction entre « SisterCrossHybridization » et « CousinCrossHybridization » qui n'a plus lieu d'être dans notre modèle générique. Seules sont stockées la longueur de l'identité entre la séquence de la sonde et la séquence du Cds, ainsi que le nombre de bases identiques sur cette longueur.

Un oligonucléotide chimérique, du type de ceux que nous utilisons dans notre nouvelle approche, est représenté par la classe **CompositeOligo**. Notre modèle permet de représenter tout type de sonde chimérique. Un CompositeOligo est composé de 2 ou plus sous-séquences (représentées par la classe **BioSequence** du MAGE-OM), ainsi que d'un ou plusieurs « linkers ».

Remarque :
Les classes **DesignGroup**, **Cds**, **Reporter** et **Biosequence** héritent toutes de la classe abstraite ***Identifiable*** du MAGE-OM qui comporte des attributs « identifier » et « name » afin de les identifier.

3.3.3 Le package DesignMethod

Ce package fournit des classes permettant de modéliser n'importe quelle méthode de design d'oligonucléotides. Sa conception est basée sur un constat : toutes les méthodes fonctionnent selon un principe général similaire. Il y a d'abord extraction d'un ou plusieurs oligos candidats du CDS. Puis on vérifie si ces oligos vérifient un certain nombre de critères, qui dépendent du contexte biologique et qui doivent être déterminés par le concepteur de l'expérience. Si un oligo vérifie l'ensemble de ces critères, il est accepté comme sonde pour le CDS considéré.

Pour concevoir le modèle, nous nous sommes inspirés du patron de conception « Strategy » [Gamma et al 1994] pour représenter le mécanisme de sélection (Figure 36). Le patron Strategy permet de définir une famille d'algorithmes, d'encapsuler chacun d'eux, et de les rendre interchangeables. La classe Context est composée d'un ou plusieurs algorithmes (classe Strategy) disponibles. La classe Strategy est une classe abstraite possédant une méthode AlgorithmInterface() qui permet d'encapsuler les différents algorithmes. Chaque algorithme est représenté par une classe ConcreteStrategy qui hérite de la classe abstraite Strategy et qui définit sa propre méthode AlgorithmInterface().

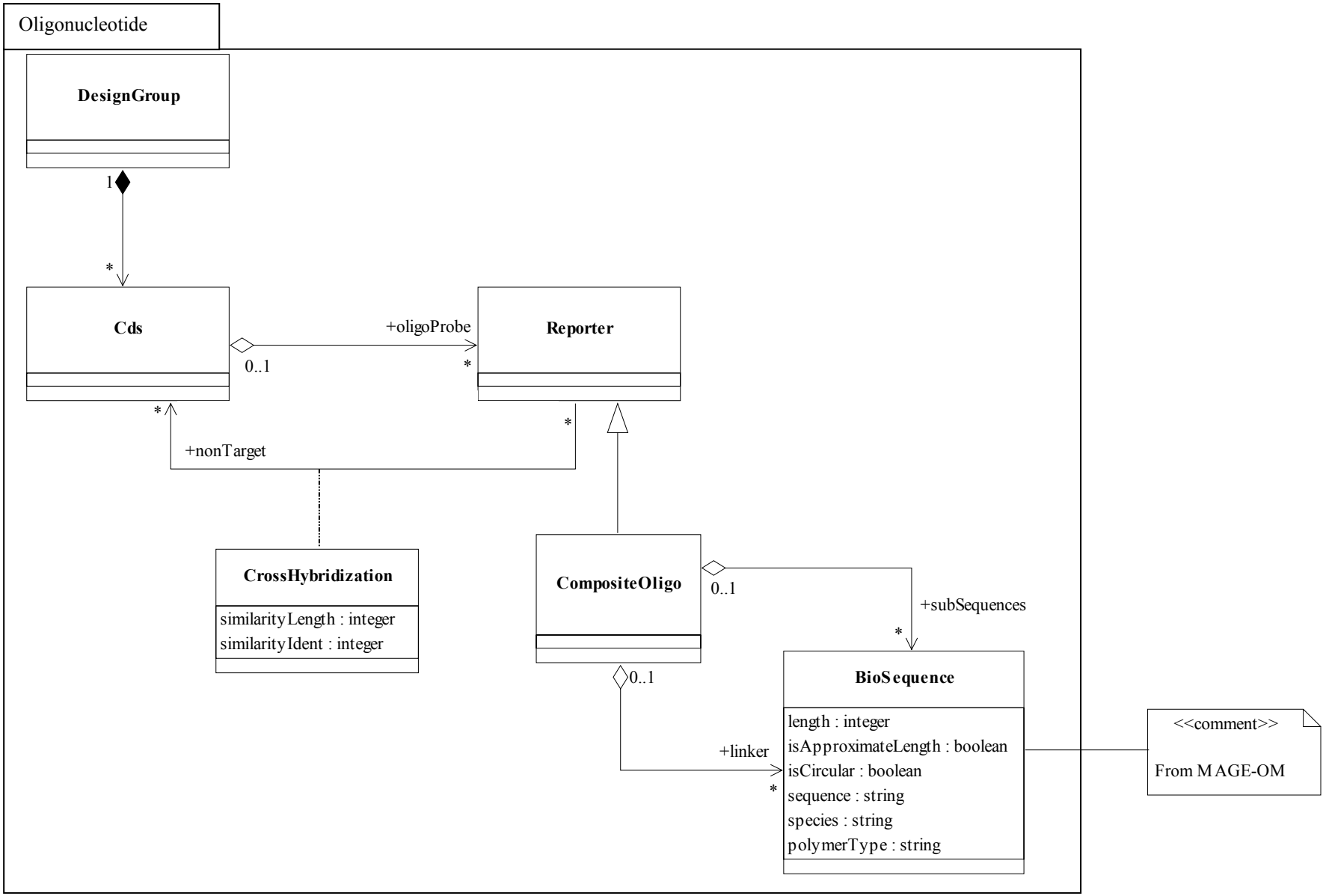


Figure 35 : Le package Oligonucleotide.

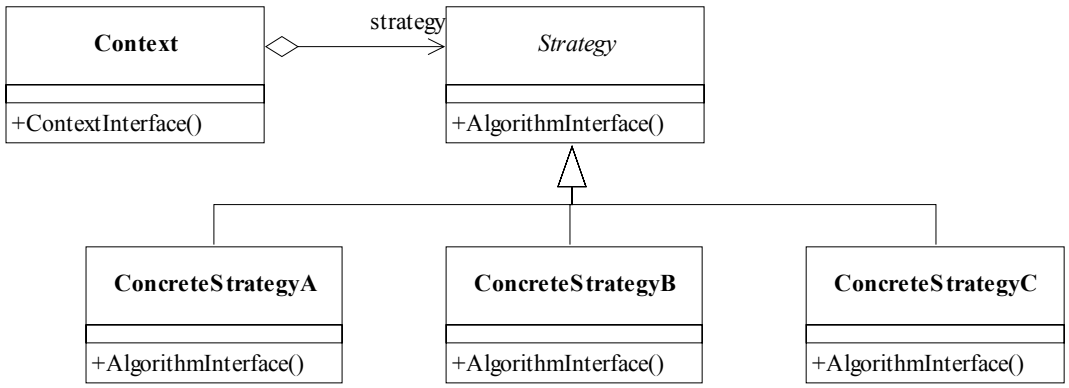


Figure 36 : Le patron de conception Strategy [Gamma et al 1994].

La Figure 37 présente le package DesignMethod. Une méthode de design de sondes sera définie par la classe **DesignContext**. Cette dernière possède une méthode « extractOligo » qui permet d'extraire un oligo d'un Cds donné. La classe DesignContext possède une référence sur un ou plusieurs tests, représentés par la classe abstraite **OligoTest**. Chaque classe héritant de cette dernière représente le test d'un critère donné sur un oligo. La méthode testOligo prend en paramètre d'entrée un oligo (Reporter) et renvoie un réel. Ce réel représente l'une des deux situations suivantes :

- Le score de l'oligo relativement au critère considéré dans le cas où l'on choisit une stratégie d'évaluation des critères de type fonction de score (voir chapitre II).
- Le réel pourra éventuellement prendre uniquement les valeurs 0 et 1 si l'on choisit une réponse de type oui/non pour l'évaluation des critères.

Les classes héritant de OligoTest représentent les critères suivants : composition en base de la séquence (**ForbidenSequenceTest**), température de fusion (**MeltingTemperatureTest**), structure secondaire (**SecondaryStructureTest**), spécificité (**CrossHybridizationTest**).

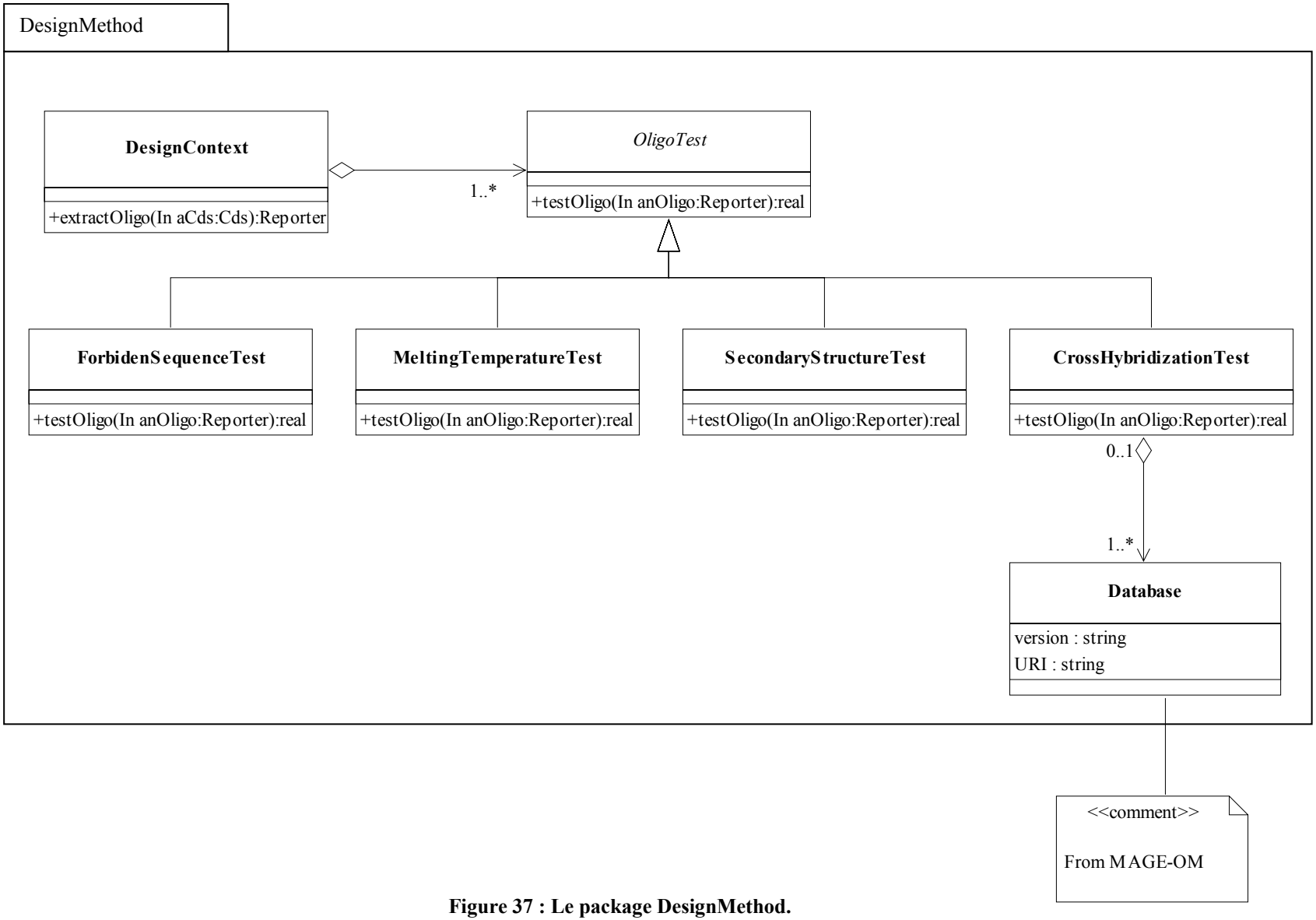


Figure 37 : Le package DesignMethod.

4 Conclusion

Dans ce chapitre, nous avons proposé des solutions aux problèmes auxquels nous étions confrontés, dans le cadre de la conception d'une biopuce destinée à étudier l'expression transcriptionnelle du parasite *Encephalitozoon cuniculi*.

Le premier problème était l'inadéquation des méthodes classiques de conception de sondes à notre contexte. En effet, l'étude d'un parasite intracellulaire obligatoire oblige à travailler avec un mélange cible complexe, composé des transcrits du parasite mais également des transcrits de l'hôte. Nous avons donc proposé une nouvelle approche sur le plan méthodologique et biologique, utilisant des sondes chimériques. La validité de cette approche a été vérifiée expérimentalement et semble prometteuse.

Le second problème était le manque de réutilisabilité des composants logiciels existants dans le domaine des puces à ADN. Des efforts importants ont été réalisés par l'OMG pour définir des standards. La MGED Society a défini un « Platform Independant Model », ainsi qu'une ontologie. Cependant, ces standards ne couvrent pas complètement le domaine de la conception d'une expérience de biopuce. C'est pourquoi nous avons proposé un PIM pour la conception de sonde, regroupant à la fois les méthodes classiques et notre nouvelle approche.

Les chapitres suivant présentent la réalisation d'outils conformes à ces propositions. Le chapitre V détaille le logiciel GoArrays, un logiciel de détermination d'oligonucléotides, réalisé à partir de notre PIM et implémentant la recherche de sondes chimériques. Nous décrivons également l'utilisation de cet outil pour la conception d'une « biopuce transcriptome » du parasite *E. cuniculi*. Le chapitre VI présente l'application de nos connaissances sur le domaine de la recherche de sondes spécifiques à un domaine très voisin : la conception de puces à ADN phylogénétiques. Un logiciel a également été réalisé pour ce problème particulier.

Chapitre V

Application : GoArrays, un logiciel de conception de sondes pour puces à ADN.

Utilisation pour la conception d'une biopuce transcriptionnelle du parasite *E. cuniculi*.

1 Introduction

Notre étude théorique sur la conception d'oligonucléotides pour puces à ADN est guidée par une problématique biologique précise : la réalisation de puces destinées à étudier le transcriptome du parasite *Encephalitozoon cuniculi*. Cette étude biologique permettra de mieux comprendre les mécanismes d'adaptation impliqués dans le cycle de développement du parasite. Nous avons vu au chapitre IV que toute la difficulté d'une telle étude résidait dans la détermination d'oligonucléotides spécifiques des gènes d'*E. cuniculi.* Nous avons proposé une nouvelle approche permettant de résoudre les problèmes posés par une telle étude.

Dans ce chapitre, nous présentons tout d'abord le logiciel GoArrays, qui implémente notre approche originale pour la détermination de sondes. D'autre part, ce logiciel est basé sur le modèle orienté objet présenté au chapitre IV, et offre tous les avantages d'une telle conception. Nous détaillons ensuite l'utilisation de ce logiciel dans le cadre du calcul des sondes pour l'ensemble du génome d'*Encephalitozoon cuniculi*, et la mise à disposition des résultats sur le Web.

2 Le logiciel GoArrays

2.1 Présentation du logiciel

GoArrays [Rimour et al 2005] est un logiciel de conception de sondes pour puces à ADN écrit en langage Java, qui implémente notre nouvelle approche utilisant des sondes chimériques (voir chapitre IV). Il est particulièrement adapté aux expériences utilisant des mélanges cibles complexes, lorsque les logiciels classiques sont mis en défaut. GoArrays dispose d'une interface graphique qui permet à l'utilisateur d'entrer les paramètres de la conception, ces paramètres sont ensuite transmis au module de design proprement dit. Ce module est une implémentation du « Platform Independant Model » présenté au chapitre IV. Le logiciel produit un fichier texte contenant la liste des sondes chimériques déterminées pour les conditions d'expérience spécifiées.

L'utilisation de GoArrays nécessite l'installation préalable des outils suivants :

- Java Runtime Environment version 1.2.2 ou supérieure.
- NCBI Blast : GoArrays utilise le programme Blast [Altschul et al 1997] pour tester la spécificité des oligonucléotides.
- GoArrays utilise le programme MFold [Zucker et al 1999] pour calculer la structure secondaire des oligonucléotides. Si l'utilisateur souhaite prendre en compte ce critère pour la conception de ses sondes (optionnel), il est donc nécessaire de disposer d'une connexion internet pour accéder au serveur MFold, ou bien d'avoir installé le logiciel MFold en local.

Pour effectuer une conception d'oligonucléotides, il est nécessaire de fournir à GoArrays deux fichiers :

- Un fichier contenant l'ensemble des CDS pour lesquels on souhaite déterminer des sondes. Ce fichier devra être au format texte FASTA.
- Une base de données contenant l'ensemble des séquences susceptibles de se trouver dans le mélange cible. Cette base de données sera utilisée pour tester la spécificité des sondes. Cette base devra être fournie sous forme d'un ensemble de fichiers au « format BLAST », c'est-à-dire l'ensemble des fichiers obtenus après utilisation de la commande « formatdb » sur un fichier FASTA.

La Figure 38 et le Tableau 11 récapitulent l'ensemble des paramètres d'entrée de l'algorithme de conception de sondes, avec leur signification précise. La Figure 39 présente l'écran principal du logiciel GoArrays. L'utilisateur peut sélectionner les critères qu'il souhaite prendre en compte pour sa conception d'oligonucléotide. Pour chacun des critères sélectionnés, le cadre de droite présente un onglet permettant de régler les options correspondant à ce critère. A titre d'exemple, la Figure 40 montre l'onglet correspondant aux paramètres de structure secondaire.

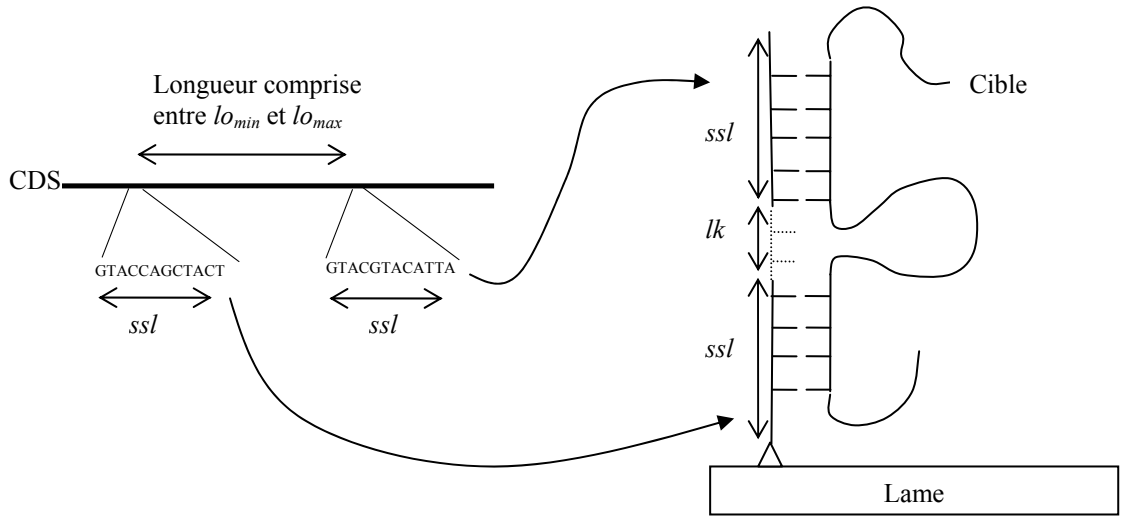


Figure 38 : Signification des paramètres *ssl, lk,* lo_{min} et lo_{max}.

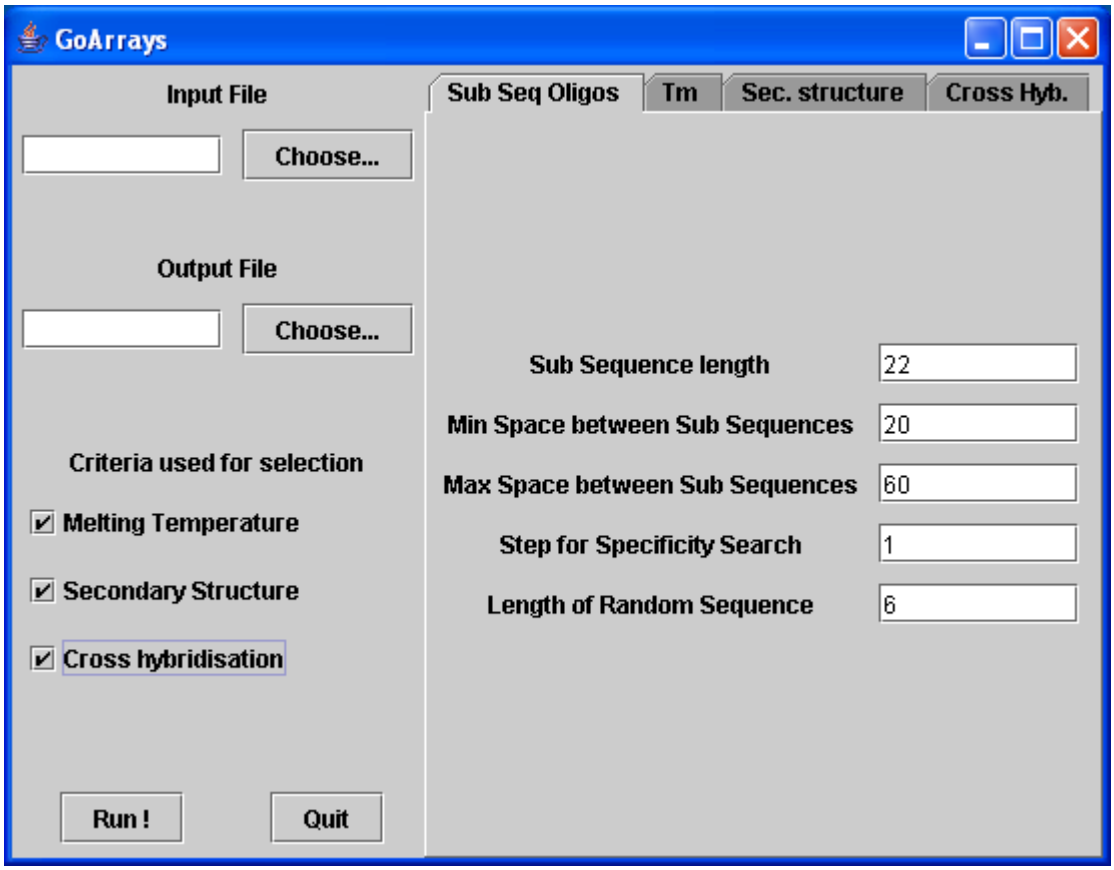


Figure 39 : Fenêtre principale du logiciel GoArrays.

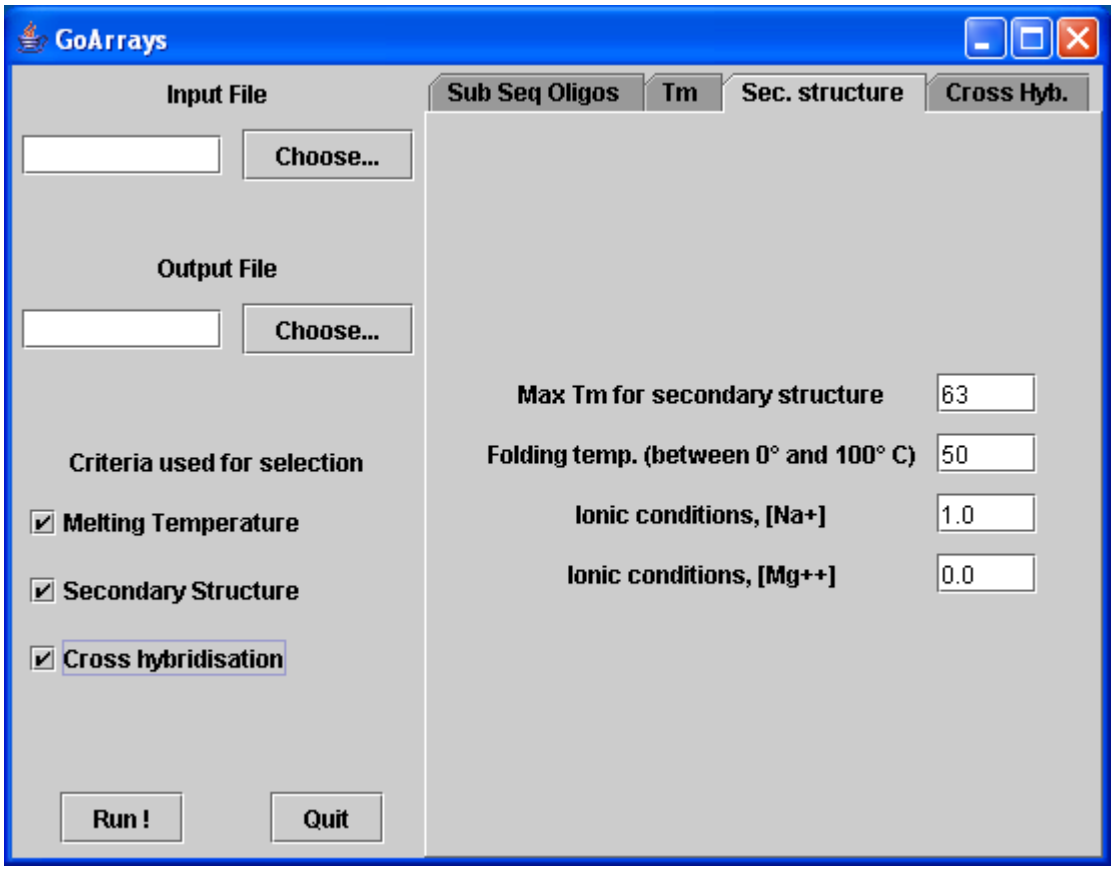


Figure 40 : L'onglet "Paramètres de structures secondaires du logiciel GoArrays".

Paramètre	Signification
Input file	Nom du fichier contenant les CDS pour lesquels on souhaiter déterminer des oligonucléotides
Output file	Nom du fichier de sortie
Subsequence length (*ssl*)	Taille en base de chaque sous séquence constituant la sonde chimérique
Min space between subsequence (lo_{min})	Longueur minimale en base de la boucle formée par la cible avec la sonde chimérique = longueur minimale entre les deux sous séquences le long du CDS
Max space between subsequence (lo_{max})	Longueur maximale en base de la boucle formée par la cible avec la sonde chimérique = longueur maximale entre les deux sous séquences le long du CDS
Step for specificity search	Pas de décalage en base le long du CDS lors de la recherche de sonde spécifique
Length of random sequence (*lk*)	Longueur en base du « linker » aléatoire
Min Tm	Température de fusion minimale acceptable pour les sondes (°C)
Max Tm	Température de fusion maximale acceptable pour les sondes (°C)
Max Tm for secondary structure	Température maximale acceptable pour une structure secondaire (°C)
Database name	Nom de la base de données contenant l'ensemble des séquences susceptibles de se trouver dans le mélange cible
Max length for identity	Permet de jouer sur la deuxième condition du critère de Kane
Folding temp.	Température du milieu pour le calcul de la structure secondaire
Ionic conditions [Na+]	Concentration en ions Na^+ pour le calcul de la structure secondaire
Ionic conditions, [Mg++]	Concentration en ions Mg^{++} pour le calcul de la structure secondaire

Tableau 11 : Les paramètres d'entrée du logiciel GoArrays.

2.2 Exemple d'exécution

Ce paragraphe présente un exemple d'exécution du logiciel GoArrays. Supposons que l'utilisateur souhaite déterminer des sondes spécifiques pour les gènes ECU_01_0960 (RIBOSOME RECYCLING FACTOR) et ECU_01_0970 (ALDOSE REDUCTASE) du parasite *Encephalitozoon cuniculi*, sachant que le mélange cible pourra contenir des ARN humains.

L'utilisateur fournit un fichier au format FASTA contenant les séquences des CDS de ces deux gènes. Il fournit également une base de données regroupant l'ensemble des séquences CDS d'*E. cuniculi* ainsi qu'un ensemble de CDS humains (par exemple la base de données UniGene [Wheeler et al 2004]). Les paramètres utilisés sont regroupés dans le Tableau 12.

Paramètre	Valeur
Subsequence length (*ssl*)	20
Min space between subsequence (lo_{min})	30
Max space between subsequence (lo_{max})	50
Step for specificity search	1
Length of random sequence (*lk*)	4
Min Tm	80
Max Tm	95
Max Tm for secondary structure	63
Database name	CDS E. cuniculi + UniGene human
Max length for identity	16
Folding temp.	50
Ionic conditions [Na+]	1.0
Ionic conditions, [Mg++]	0.0

Tableau 12 : Paramètres utilisés pour l'exemple d'exécution.

Le logiciel calcule alors une sonde spécifique pour chacun des deux gènes et fournit le fichier de sortie suivant :

```
>01_0960
Tm=86
Pos=181  237
TATCGATCGAGCAGCTGCAAACGTAACTCCAAGAGGATCGAGAG

>01_0970
Tm=88
Pos=128  187
AAGTATGGATGCGCCCCGTCAACGTGTGATTCCGAAAAGCAGAT
```

Pour chaque gène sont indiquées la température de fusion de la sonde, la position de chaque sous séquence (indiquées en gras) le long du CDS, ainsi que la séquence complète de la sonde.

2.3 Conception

2.3.1 Modèle

GoArrays implémente un « Platorm Specific Model », lui-même issu du « Platform Independant Model » pour la conception d'oligonucléotides présenté au chapitre IV. La Figure 41 présente les relations entre les différents packages utilisés par GoArrays. Le logiciel utilise deux librairies externes, l'une pour lancer le programme BLAST, et l'autre pour analyser les résultats de ce même programme BLAST. Ces librairies sont décrites dans le paragraphe suivant.

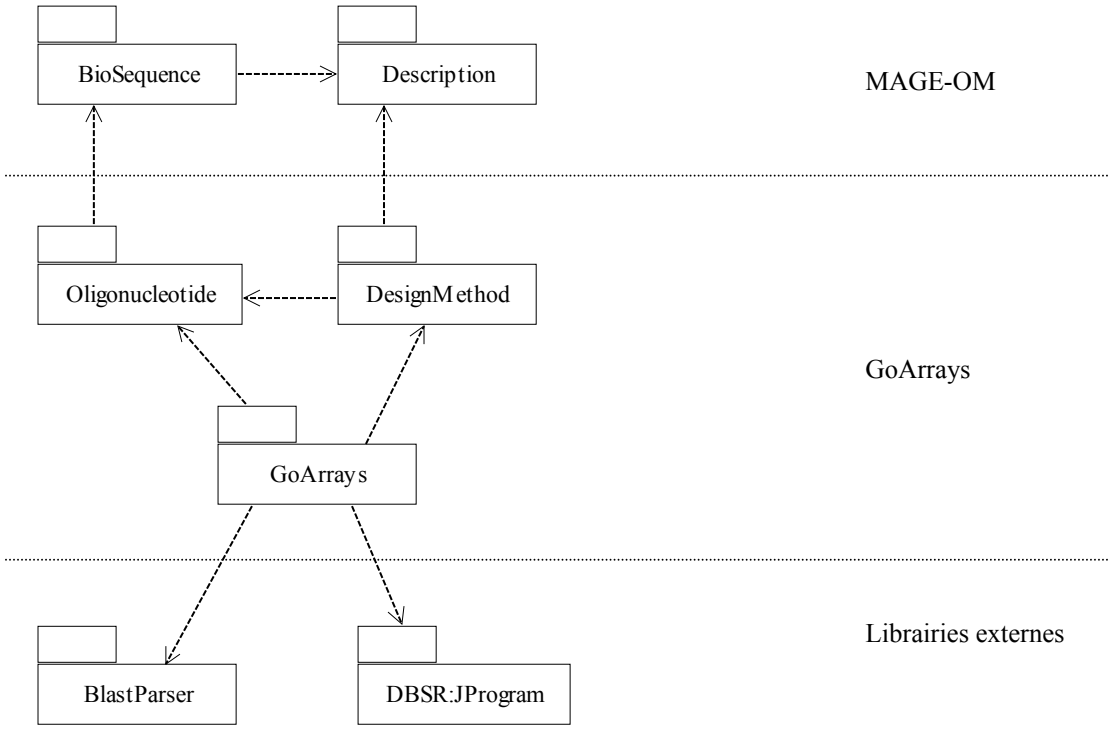


Figure 41 : Dépendances entre les packages du logiciel GoArrays.

2.3.2 Outils externes utilisés

Le lancement de BLAST par GoArrays utilise une librairie nommée **JProgram** développée par le « Duke Bioinformatics Shared Ressource » (DBSR[18]). Cette librairie permet de lancer des exécutables couramment utilisés en bioinformatique tels BLAST ou Clustalw à partir d'un programme Java. L'appel à ces programmes est encapsulé dans des méthodes de classes et les « ressources » nécessaires à leur exécution (exécutables, bases de données) sont spécifiées par l'utilisateur dans un fichier XML. Ceci permet par exemple de pouvoir utiliser GoArrays avec un autre BLAST que celui du NCBI.

[18] Centre de bioinformatique de Duke University :
http://www.dbsr.duke.edu/support/softwaredev/libraries/java/jprogram/

L'utilisateur de GoArrays spécifie les ressources nécessaires à l'utilisation de BLAST dans un fichier XML de ce type :

```
<resources>
      <resourceSet>
            <name>NCBI Blast</name>
            <resource>
                  <type>executable</type>
                  <name>blastall</name>
            <value>c:\Blast\blastall</value>
            </resource>
            <resource>
                  <type>database</type>
                  <name>ecoli.nt</name>
                  <value>c:\Blast\db\ecoli.nt</value>
            </resource>
      </resourceSet>
</resources>
```

L'élément **RessourceSet** représente la « ressource », c'est-à-dire le logiciel externe qui est lancé à partir du programme Java. On définit à l'intérieur de cet élément le chemin vers l'exécutable ainsi que la liste des bases de données que l'on souhaite utiliser.

La seconde librairie utilisée par GoArrays est **Java Blast Parser** développée par le Harvard Institute of Proteomics[19]. Il s'agit d'un analyseur de fichier de sortie BLAST écrit en Java. Il permet de récupérer aisément les informations contenues dans un tel fichier en utilisant un analyseur grammatical.

2.3.3 Algorithme

Etant donné un CDS, le programme extrait les oligonucléotides candidats en partant de l'extrémité 3'. L'algorithme de recherche de sonde est présenté Figure 42 et Figure 43. Il commence par chercher une première sous-séquence spécifique en déplaçant une fenêtre de longueur *ssl* le long du CDS. Puis il recherche la seconde sous-séquence spécifique en tenant compte des paramètres lo_{min} et lo_{max}. Quand les deux sous-séquences ont été trouvées, la séquence aléatoire « linker » est ajoutée entre les deux. Cette dernière est modifiée (nouveau tirage aléatoire) jusqu'à ce que la sonde chimérique globale vérifie le critère de spécificité, car l'introduction du linker peut introduire de nouvelles hybridations croisées. Enfin, les autres critères définis par l'utilisateur sont vérifiés.

Afin de vérifier les différents critères, les méthodes suivantes sont utilisées :

- Spécificité de la sonde : on effectue un BLAST de la séquence contre la base de données contenant tous les transcrits susceptibles de se trouver dans le mélange cible. Si on trouve un alignement contenant 15 bases ou plus consécutives identiques, ou si un alignement présente plus de 75% de similarité, la séquence est considérée comme non spécifique.
- Température de fusion : la méthode utilisée est celle des plus proches voisins [SantaLucia 1998].
- Structure secondaire : GoArrays appelle le programme MFold [Zucker et al 1999] avec les paramètres définis par l'utilisateur.

[19] http://www.hip.harvard.edu/informatics/

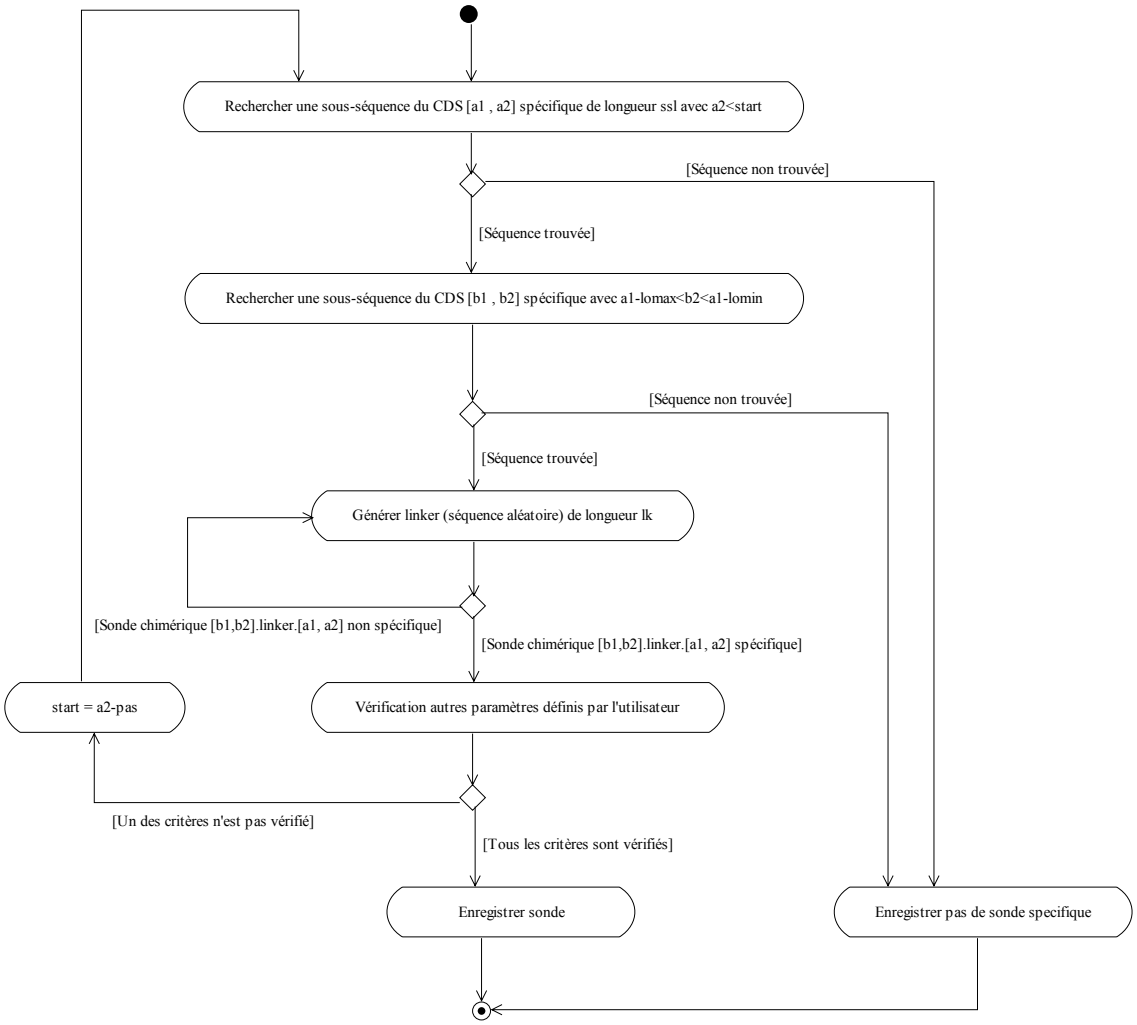


Figure 42 : diagramme d'activité représentant l'algorithme de recherche de sonde chimérique.

2.3.4 Bilan

Le logiciel GoArrays permet donc de déterminer des sondes conformément à notre nouvelle approche, proposée et validée expérimentalement au chapitre IV. Pour sa conception, nous avons suivi une démarche s'inspirant de l'approche MDA. Ainsi, la maintenance du code et l'évolution du logiciel seront grandement facilités. Par exemple, on peut aisément rajouter un critère de test pour les oligonucléotides, il suffira d'implémenter une nouvelle classe héritant de la classe OligoTest dans le package DesignMethod.

Dans le paragraphe suivant, nous présentons l'utilisation de GoArrays pour une application biologique réelle : la conception d'une biopuce visant à étudier le transcriptome du parasite *Encephalitozoon cuniculi*.

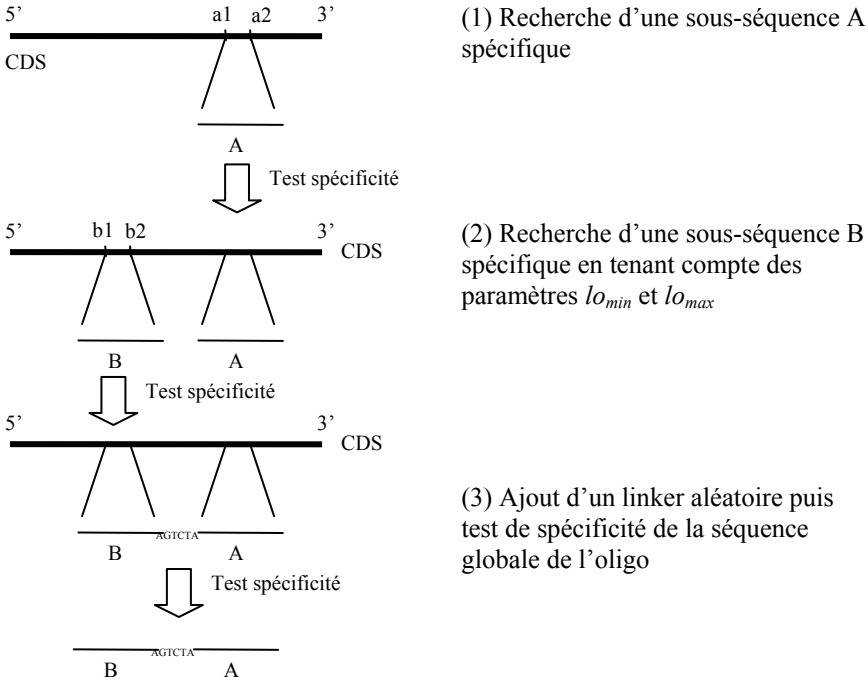


Figure 43 : Schéma de la recherche d'une sonde chimérique spécifique.

3 Application à l'étude de l'expression transcriptionnelle du parasite Encephalitozoon cuniculi

Rappelons que l'objectif de l'équipe Génomique Intégrée des Interactions Microbiennes est de concevoir une puce à ADN spécifique du pathogène *Encephalitozoon cuniculi* pour identifier les différentes classes de gènes exprimés au cours du développement parasitaire. Cette puce doit donc cibler l'ensemble du transcriptome du parasite et les sondes doivent être les plus spécifiques possibles. Le design des oligonucléotides doit être réalisé en tenant compte du fait que le mélange cible contiendra un mélange des transcrits du parasite et de transcrits humains. Nous avons vu que les logiciels de conception d'oligonucléotides existants donnaient des résultats décevants dans ces conditions d'expérience et nous avons proposé une nouvelle approche pour la conception de sondes. Ce paragraphe présente la conception de cette biopuce « transcriptome » pour *E. cuniculi*.

3.1 Calcul des oligos

Afin de comparer notre nouvelle approche à une méthode de conception classique, nous avons décidé de déposer deux types de sondes sur la puce :

- Des sondes calculées avec le logiciel Oligoarray [Rouillard et al 2002].
- Des sondes calculées avec notre logiciel GoArrays.

Des sondes ont été calculées pour cibler les 1995 CDS identifiés *d'Encephalitozoon cuniculi*. Afin de mettre en évidence des régulations qui seraient dues à des expressions d'ARN chevauchant ou anti-sens, nous avons décidé de déterminer également des sondes pour les 1995 séquences anti-sens (séquence inverse complémentaire) des CDS.

3.1.1 Calcul des « sondes OligoArray »

Pour chaque gène d'*E. cuniculi*, trois sondes ont été calculées à l'aide du logiciel OligoArray. Nous souhaitions que ces trois sondes soient à peu de choses près réparties le long du CDS, c'est à dire qu'elles ne soient pas toutes proches de l'extrémité 3' ou 5' par exemple. Comme OligoArray ne permet pas à l'utilisateur de contrôler la position des oligos calculés, nous avons découpé chaque CDS en trois séquences de longueur égale et chacune d'entre elles a été soumise au logiciel (Figure 44).

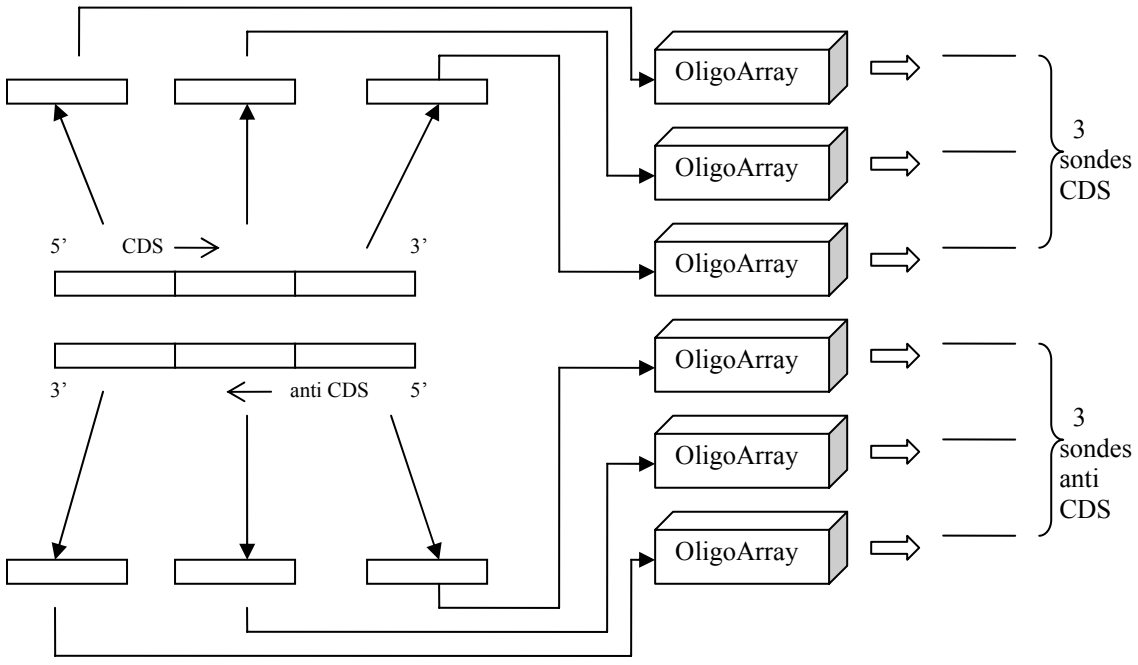


Figure 44 : Calcul des sondes pour un gène d'E. cuniculi avec le logiciel OligoArray.

Le même travail a été réalisé pour les séquences inverses complémentaires des CDS afin de disposer également de trois sondes pour la région anti-sens correspondante.

Le Tableau 13 résume les paramètres utilisés pour calculer ces sondes. Au final, nous disposons de 6 « sondes OligoArray » par gène, soit 11935 sondes au total (pour certains gènes, le logiciel n'arrive pas à déterminer de sonde).

Nom du paramètre	Signification	Valeur
blastDB	Base de donnée spécificité	CDS E. cuniculi + UniGene human
oligoLength	Longueur des oligos	50
Distance 5' - stop	Limite position des oligos	10000
TmRange	Température de fusion	80
TmRange(+/-)	-	95
maxTm	Tm de structure secondaire	63
mxNbOligo	Nombre d'oligos par gène	1
listProhibited	Séquences interdites	AAAAA TTTTT GGGGG CCCCC
linker5prime	-	-
linker3prime	-	-

Tableau 13 : Paramètres utilisés pour le calcul des sondes avec le logiciel OligoArray.

3.1.2 Calcul des « sondes GoArrays »

Une sonde chimérique a été calculée à l'aide de notre logiciel GoArrays pour chaque gène d'*E. cuniculi*, à l'exception des gènes situés aux extrémités des chromosomes (duplications parfaites). En effet, le génome d'*E. cuniculi* présente des ensembles de gènes dont les séquences sont extrêmement proches voire identiques. Ces gènes sont regroupés dans des zones situées aux extrémités des chromosomes. Ces gènes dupliqués ont mis en échec notre algorithme de recherche de sondes spécifiques, c'est pourquoi nous n'avons pas calculé de sonde « GoArrays » pour une dizaine de CDS au début et à la fin de chaque chromosome.

Une sonde chimérique a également été calculée pour la région anti-sens correspondant à chaque CDS.

Le Tableau 14 résume les paramètres utilisés dans le logiciel. Au final, nous disposons de 2 sondes chimériques par gène, soit 3429 sondes au total.

Nom du paramètre	Valeur
Subsequence length	20
Min space between subsequence	30
Max space between subsequence	50
Step for specificity search	1
Length of random sequence	4
Min Tm	80
Max Tm	95
Max Tm for secondary structure	63
Database name	CDS E. cuniculi + UniGene human
Max length for identity	16
Folding temp.	50
Ionic conditions [Na+]	1.0
Ionic conditions, [Mg++]	0.0

Tableau 14 : Paramètres utilisés pour le calcul les sondes avec le logiciel GoArrays.

(pour la signification des paramètres, voir paragraphe 2.1)

3.2 Base de données et interface Web

L'ensemble des sondes calculées est stocké dans une base de données MySQL et accessible via une interface Web :
http://fc.isima.fr/~rimour/oligo/

La Figure 45 présente la page d'accueil avec le formulaire permettant de rechercher rapidement un CDS d'*E. cuniculi* par son identifiant. La Figure 46 montre les séquences d'oligonucléotides calculées pour un CDS donné : 3 sondes de type OligoArray avec leur position sur le CDS, ainsi qu'une sonde de type GoArrays. Il est possible de visualiser les hybridations croisées potentielles pour chaque sonde non spécifique.

Il est également possible de naviguer dans l'ensemble des gènes, classés par chromosome (Figure 47).

L'ensemble des pages Web est réalisé en XHTML/CSS, ainsi qu'en PHP pour les accès à la base de données MySQL.

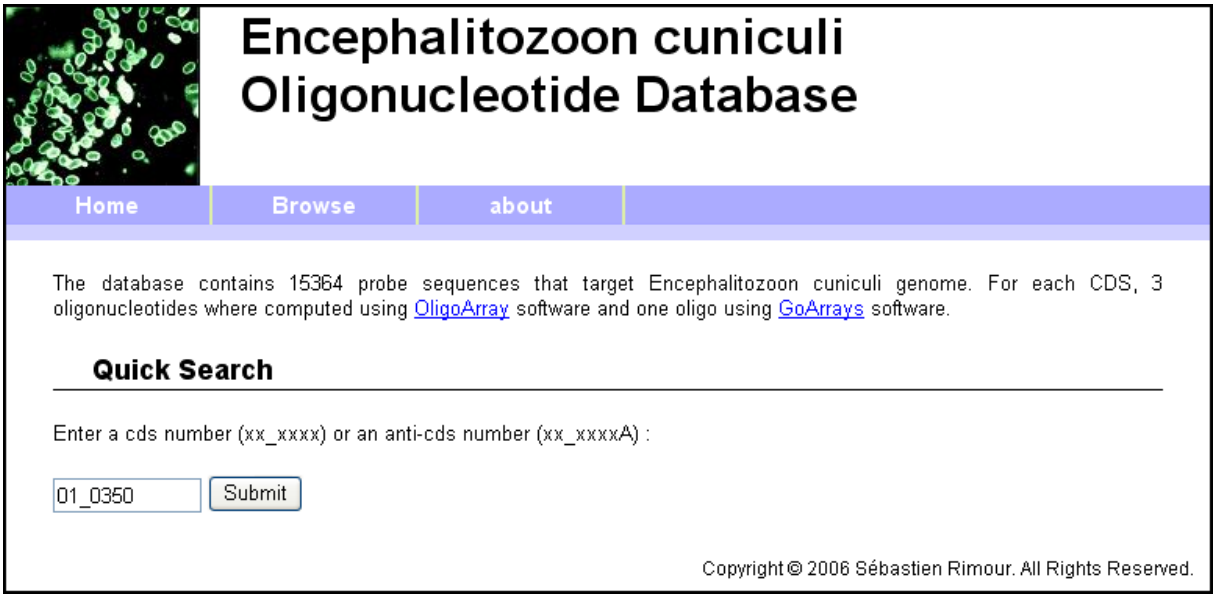


Figure 45 : Page d'accueil de la base de données d'oligonucléotides.

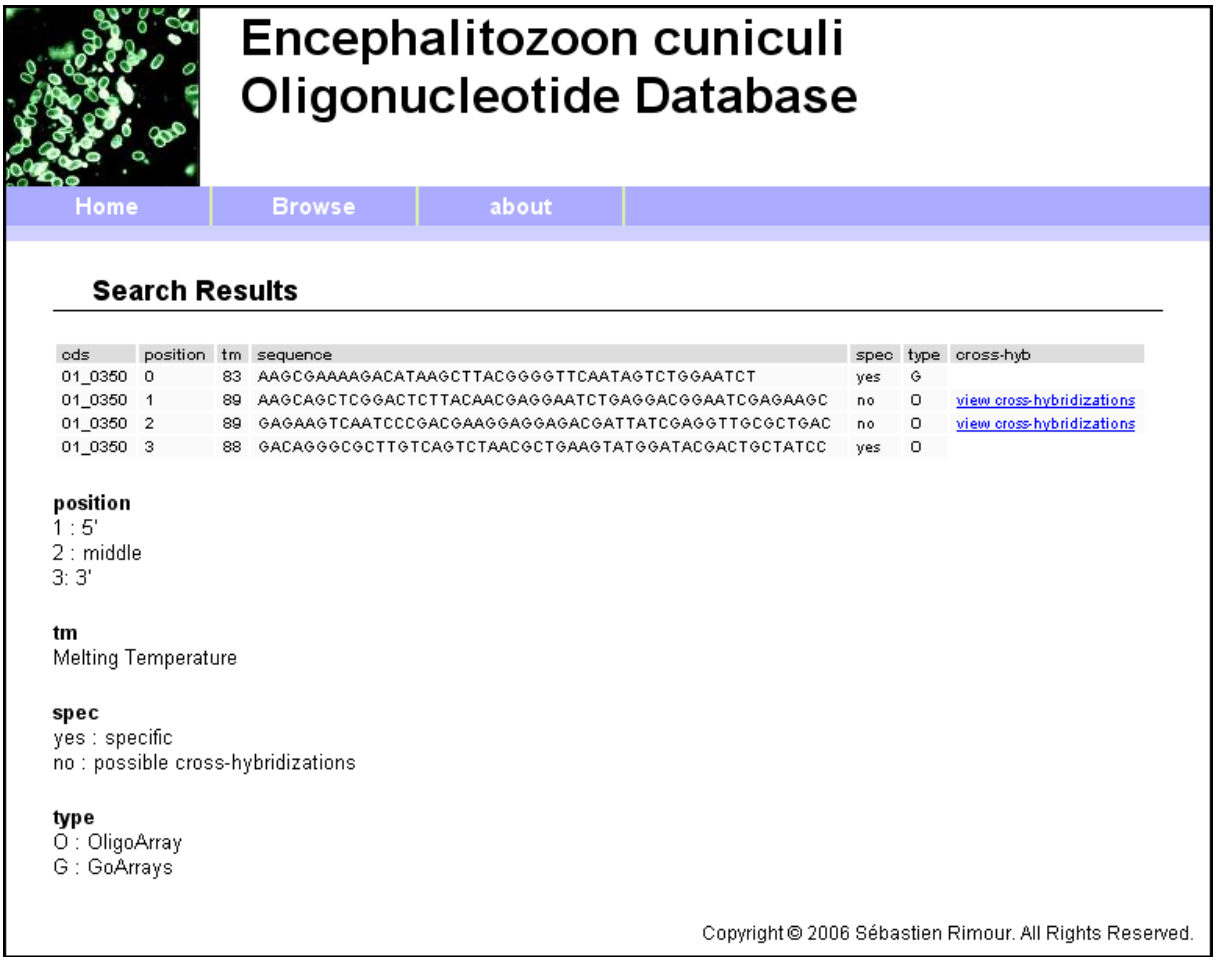


Figure 46 : Affichage des oligonucléotides pour un CDS donné.

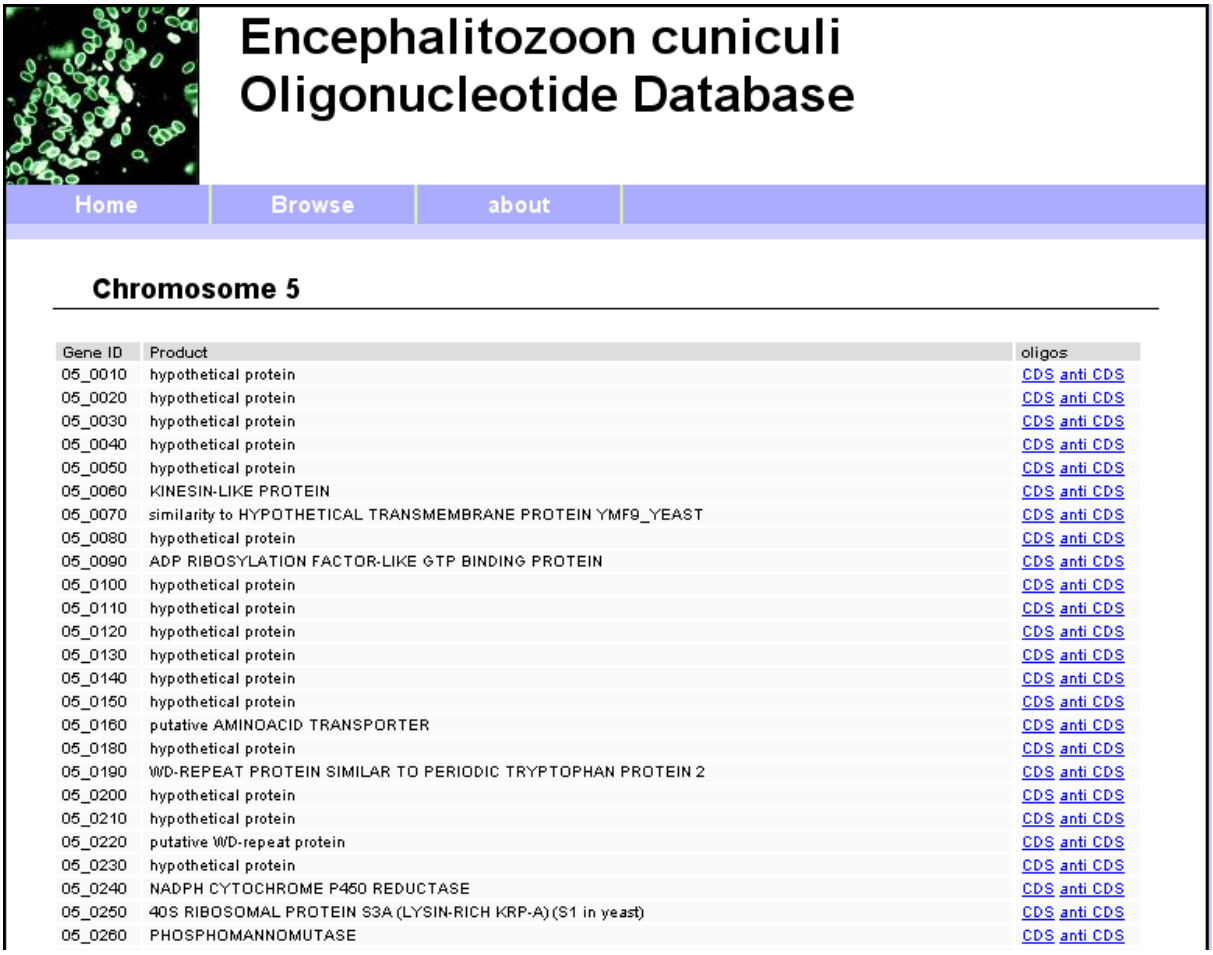


Figure 47 : Navigation parmi les CDS d'un chromosome.

4 Conclusion

Dans ce chapitre, nous avons vu l'application directe des propositions faites au chapitre IV. Le logiciel GoArrays a été développé à partir du "Platform Independant Model" de conception de sondes. Cette approche permettra une bonne maintenance du code et facilitera son évolution. De plus, GoArrays calcule les oligonucléotides selon notre nouvelle approche utilisant des sondes composites. Il a été utilisé pour déterminer les sondes ciblant l'ensemble des gènes du parasite *Encephalitozoon cuniculi*, modèle d'étude des biologistes du LBP avec qui nous collaborons.

Un deuxième axe de recherche de l'équipe du LBP est la réalisation de puces à ADN dites « phylogénétiques ». Ces dernières ne sont plus destinées à l'étude de l'expression transcriptionnelle mais à l'identification d'organismes dans un environnement complexe. Le

problème de conception de sondes pour de telles biopuces présente des similitudes avec celui des biopuces transcriptionnelles. Nous avons souhaité tirer profit de l'expérience acquise dans la conception de sondes afin de proposer de nouvelles approches dans le domaine des puces phylogénétiques. C'est l'objet du chapitre suivant, dans lequel nous présentons un algorithme original de conception de sondes ainsi qu'un logiciel accessible via Internet.

Chapitre VI

Application : PhylArray, un logiciel de conception de sondes pour puces à ADN phylogénétiques

1 Le cas particulier des puces phylogénétiques

1.1 Contexte biologique

L'ensemble des méthodes et algorithmes présentés dans le chapitre II concernait essentiellement la conception de sondes pour des biopuces à ADN destinées à la mesure de l'expression transcriptionnelle. La deuxième problématique biologique ayant guidé les recherches présentées dans cette thèse est la mise au point de biopuces destinées à suivre l'évolution des communautés bactériennes lors d'un processus de biorémédiation (voir chapitre I). Il s'agit, à l'aide d'une puce à ADN, de déterminer quels sont les micro-organismes présents dans un milieu donné. Sur cette puce, une espèce sera représentée par un oligonucléotide, ou un groupe d'oligonucléotides, et la mesure de la fluorescence des spots correspondants indiquera la présence ou non de l'espèce dans le milieu. Pour réaliser de telles expériences, il est nécessaire de cibler avec les sondes une molécule qui puisse constituer une véritable « carte d'identité moléculaire » pour le monde vivant.

L'ARN ribosomique, molécule présente chez tous les organismes, participe à la structure des ribosomes. Il existe une importante variabilité de la séquence de cette molécule suivant les organismes, et c'est donc l'information contenue dans le gène codant pour cette molécule (l'ADNr 16S) qui va être utilisé comme identificateur.

La discipline de la biologie qui étudie la classification des êtres vivants en fonction de leur évolution est la phylogénie. Du fait de la diversité des êtres vivants rencontrés, ce classement est très difficile à effectuer et en constante évolution, au fur et à mesure que de nouvelles études et de nouvelles découvertes sont réalisées. La classification des bactéries entre elles reposait initialement sur plusieurs types d'observations et d'études. Elles peuvent ainsi être classées et donc identifiées en fonction de leur morphologie (microscopique et macroscopique), de leur mobilité, de la température de croissance, du type respiratoire, des besoins nutritionnels, etc. Aujourd'hui, avec l'explosion des nouvelles techniques de biologie moléculaire et du séquençage de l'ARNr 16S, la nomenclature bactérienne a été profondément modifiée, au gré des connaissances génétiques.

Les études dites phylogénétiques permettent d'établir la distance génétique qu'il existe entre les espèces, cette distance étant basée sur une mesure de similarité entre certaines séquences nucléiques (comme l'ARNr). En effet, plus les séquences nucléiques seront proches, plus les espèces étudiées seront voisines. A l'inverse, si les séquences nucléiques sont moins homologues, on pourra considérer que les espèces en questions sont plus éloignées.

Dans la classification du vivant, les organismes procaryotes (*Procaryotae*) regroupent les organismes unicellulaires ne présentant pas de noyau individualisé, c'est à dire les Bactéries et les Archaebactéries. Les Eucaryotes (*Eucarya*) regroupent quant à eux l'ensemble des organismes unicellulaires ou multicellulaire à noyau individualisé. Le règne (*Procaryotae*) est le premier niveau de classification. Vient ensuite le domaine (*Bacteria*), le phylum, la classe,

l'ordre, la famille, le genre, et l'espèce. Cette dernière constitue l'unité de classification. Toutefois, il est souvent nécessaire de subdiviser une espèce en différentes sous-espèces (subspecies).

Voici par exemple la position de la bactérie bien connue *Escherichia coli* dans l'arbre du vivant :

règne : *Procaryotae*

domaine : *Bacteria*

phylum : *Proteobacteria*

classe : *Gammaproteobacteria*

ordre : *Enterobacteriales*

famille : *Enterobacteriaceae*

genre : *Escherichia*

espèce : *Escherichia coli*

Le monde bactérien présente une diversité considérable et on estime que des milliers de micro-organismes différents peuplent la plupart des environnements. Cependant, la plupart d'entre eux n'ont pas encore été décrits car il n'est pas encore possible de les cultiver en milieu artificiel [Ward et al 1992, Amann et al 1995]. La séquence du gène codant pour l'ARNr 16S a largement été utilisée pour identifier et classifier la diversité des communautés bactériennes [Woese et al 1975, Fox et al 1980]. L'utilisation de la technologie des puces à ADN permet maintenant de suivre simultanément l'évolution de plusieurs milliers de micro-organismes différents. Ces puces sont dites « phylogénétiques ».

1.2 Le problème de la conception d'oligonucléotides pour puces phylogénétiques

Le problème de la conception de sondes oligonucléotidiques pour les puces phylogénétiques présente de nombreuses similitudes avec celui de la conception de sondes pour puces transcriptomiques. L'objectif est toujours le même : identifier une sonde spécifique s'hybridant de manière optimale avec le transcrit correspondant. Cette sonde devra être une sous séquence du transcrit ciblé et devra satisfaire au mieux à un certain nombre de critères. Parmi les critères à satisfaire, on retrouvera la température de fusion, la composition en base de la séquence et la structure secondaire.

La principale particularité de ces sondes pour puces phylogénétiques concerne la **spécificité** de la séquence. En effet, **une sonde devra posséder une séquence spécifique du transcrit**

ciblé (la séquence d'ARNr 16S), **non pas par rapport aux autres gènes du même organisme, mais par rapport à toutes les séquences d'ARNr 16S connues de tous les autres organismes susceptibles de se trouver dans le mélange cible.** La base de données servant à tester la spécificité devra donc contenir l'ensemble des séquences d'ARNr 16S des organismes susceptibles de se trouver dans le mélange cible.

1.3 Méthodes et logiciels existants

Il existe beaucoup moins d'études pour ce problème que pour le problème classique de la conception de sondes pour puces transcriptomiques.

Les premières études ne recherchaient pas la séquence de la sonde sur l'ensemble de la molécule d'ARNr mais se concentraient sur une zone bien connue d'une centaine de bases [Wilson et al 2002]. Un grand nombre d'oligos courts (20mers) répartis sur cette zone étaient alors choisis, cette méthode est connue sous le nom d' « oligonucleotide fingerprinting ». Plutôt que de chercher à déterminer des séquences spécifiques, c'est le recoupement des résultats obtenus pour chaque oligo qui permettait d'identifier les organismes. Des algorithmes ont été proposés pour minimiser le nombre de sondes nécessaires par organisme [Borneman et al 2001].

Zhang et al [2002] proposent un algorithme permettant de déterminer dans quelle mesure un oligonucléotide est spécifique d'un organisme ou d'un groupe d'organismes donnés. La recherche d'une sonde spécifique consiste alors à parcourir l'ensemble des oligos potentiels et à calculer leur « qualité de signature » Q_s.

Borneman et al [2001] utilisent une approche assez originale : ils modélisent la question sous forme d'un problème d'optimisation et utilisent les algorithmes du recuit simulé et de la relaxation Lagrangienne pour le résoudre. L'inconvénient est que ces travaux sont basés sur les premières expériences d' « oligonucléotide fingerprinting » et que les séquences déterminées sont très courtes (10-20 bases) et donc peu sensibles.

Certaines méthodes sont basées sur le regroupement hiérarchique (ou clustering) de l'ensemble des séquences d'ARNr connues [DeSantis et al 2003]. On considère alors que les organismes appartenant à un même cluster ne pourront pas être différenciés car leurs séquences d'ARNr 16S sont trop proches. Les algorithmes cherchent alors à maximiser la distance avec les autres clusters. Les méthodes diffèrent par la mesure de distance (basée sur la similarité) utilisée entre les séquences nucléiques [Hazelhurst et al 2003].

Enfin, il existe deux logiciels principaux, librement téléchargeables, permettant de concevoir des sondes pour puces à ADN phylogénétiques. Le premier est PRIMROSE [Ashelford et al 2002] : il s'agit d'un logiciel écrit en Java qui utilise la principale base de données de séquences d'ARNr (nommée RDP, voir paragraphe 2.2). Le processus de sélection de sonde avec PRIMROSE est le suivant :

1. Choix de la base de données d'ARNr utilisée. Il s'agit de RDP par défaut mais il est possible d'exclure certaines séquences de la base ou d'utiliser une base complètement différente.

2. Sélection du groupe d'organismes à cibler. L'objectif du programme est alors de déterminer une ou plusieurs sondes spécifiques de ce groupe.

3. Sélection, à l'intérieur du groupe d'organismes à cibler des séquences d'ARNr utilisées pour la recherche de sondes.

4. Recherche des sondes et tests de spécificité

Le principal inconvénient de ce logiciel est que l'étape 3 est manuelle : l'utilisateur doit cocher les séquences à prendre en compte pour la recherche de sondes.

Le second logiciel est ARB [Ludwig et al 2004], écrit en C++. Il s'agit d'un logiciel très complet de manipulation de données phylogénétiques. L'une de ses fonctions a été utilisée pour calculer des sondes spécifiques sur une base de données contenant 20000 séquences d'ARNr. Cette base de données de sondes précalculées est accessible via un client Java.

2 Problème

2.1 Introduction

Le problème est de déterminer la séquence d'une sonde oligonucléotidique permettant d'identifier un organisme ou un groupe d'organisme à partir de la molécule d'ARNr 16S.

Exemple :

Le gène codant l'ARNr 16S de la bactérie *Micrococcus luteus* a été séquencé, et a une longueur de 1430 bases :

```
>S000004845 Micrococcus luteus; D7; AJ409095

CATGCAAGTCGAACGATGAAGCCCAGNNTGCTGGTGGATTAATGGCGAACGGGTGAGTAACACGTGAGTNACCTGCCCTTAA
CTCTGGGATAAGCCTGGGAAACTGGGTCTAATACCGGATAGGAGCGTCCACCGCATGGTGGGTGTTGGAAAGATTTATCGGT
TTTGGATGGACTCGCGGCCTATCAGCTTGTTGGTGAGGTAATGGCTCACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGT
GACCGGCCACACTGGGACTGAGACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGAAAGCC
TGATGCAGCGACGCCGCGTGAGGGATGACGGCCTTCGGGTTGTAAACCTCTTTCAGTAGGGAAGAAGCGAAAGTGACGGTAC
CTGCAGAAGAAGCACCGGCTAACTACGTGCCAGCAGCCGCGGTAATACGTAGGGTGCGAGCGTTATCCGGAATTATTGGGCG
TAAAGAGCTCGTAGGCGGTTTGTCGCGTCTGTCGTGAAAGTCCGGGGCTTAACCCCGGATCTGCGGTGGGTACGGGCAGACT
AGAGTGCAGTAGGGGAGACTGGAATTCCTGGTGTAGCGGTGGAATGCGCAGATATCAGGAGGAACACCGATGGCGAAGGCAG
GTCTCTGGGCTGTAACTGACGCTGAGGAGCGAAAGCATGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCATGCCGTAAA
CGTTGGGCACTAGGTGTGGGGACCATTCCACGGTTTCCGCGCCGCAGCTAACGCATTAAGTGCCCCGCCTGGGGAGTACGGC
CGCAAGSTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGCGGACATGCGGATTAATTCGATGCAACGCGAAGAACC
TTACCAAGGCTTGACATGTTCTCGATCGCCGTAGAGATACGGTTTCCCCTTTGGGGCGGGTTCACAGGTGGTGCATGGTTGT
CGTCAGCTCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTCGTTCCATGTTGCCAGCACGTAATGGTGG
GGACTCATGGAAGACTGCCGGGGTCAACTCGGAGGAAGGTGAGGACGACGTCAAATCATCATGCCCCTTATGTCTTGGGCTT
CACGCATGCTACAATGGCCGGTACAATGGGTTGCGATACTGTGAGGTGGAGCTAATCCCAAAAAGCCGGTCTCAGTTCGGAT
TGGGGTCTGCAACTCGACCCCATGAAGTCGGAGTCGCTAGTAATCGCAGATCAGCAACGCTGCGGTGAATACGTTCCCGGGC
CTTGTACACACCGCCCGTCAAGTCACGAAAGTTGGTAACACCCGAAGCCGGTGGCCTAACCCTTGTGGGGGGAGCCGTCGAA
GGTGGGACCAGCGATTGGGACTAAGTNGTAACAAGG
```

Notre problème est similaire à celui de la détermination de sondes pour des puces transcriptomes classiques. Il s'agit de déterminer une sous-séquence de la molécule d'ARNr 16S qui ne puisse pas s'hybrider avec d'autres séquences susceptibles de se trouver dans le mélange cible. Ainsi, si l'on souhaite identifier la bactérie dans un environnement totalement inconnu, cette sous-séquence devra vérifier le critère de Kane par rapport à l'ensemble des molécules d'ARNr 16S connues. Pour être complètement opérationnelle, la sonde devra en plus vérifier un maximum de critère parmi les critères classiques de la conception d'oligonucléotides (Température de fusion, pas de structure secondaire stable, …).

L'équipe Génomique Intégrée des Interactions Microbiennes du Laboratoire de Biologie des Protistes souhaitait définir une contrainte supplémentaire dans la détermination des sondes oligonucléotidiques. En effet, actuellement, la plupart des microorganismes du sol étant encore non identifiés, l'équipe ne désirait pas utiliser uniquement des sondes complémentaires spécifiques d'espèces connues, ce qui rendrait la biopuce trop spécifique sans possibilité réelle d'identifier de nouvelles espèces actives. La stratégie est donc de désigner à terme, des sondes capables de cibler des groupes de micro-organismes au niveau de l'ordre, de la famille et du genre. La biopuce permettra de connaître les groupes qui varient lors de la bioremédiation et donc ceux qui ont potentiellement un rôle dans ces processus (populations stables ou en augmentation).

Les sondes devront donc être capable, si le biologiste le souhaite, d'identifier un groupe d'organisme (typiquement un genre) et non plus une espèce donnée. Cette contrainte impose de sélectionner des sondes dont la séquence est à la fois commune au groupe ciblé, mais également spécifiques par rapport à l'ensemble des autres séquences d'ARNr 16S connues (Figure 48).

2.2 Les données génétiques disponibles

La molécule d'ARNr 16S a été séquencée chez un très grand nombre de micro-organismes et est largement utilisée pour des études phylogénétiques. Ainsi, il est courant d'identifier la position dans l'arbre phylogénétique d'une bactérie inconnue en amplifiant et séquencant sa molécule d'ARNr 16S [Zhang et al 2002]. Les principales sources de séquences d'ARNr sont d'une part les grandes banques de données généralistes telles GenBank [Benson et al 2006] ou l'EMBL [Cochrane et al 2006] et d'autre part des bases de données de taille plus réduite contenant exclusivement des séquences d'ARNr, comme nous le verrons par la suite.

Les grandes banques de données généralistes sont les plus fournies en séquences d'ARN ribosomique de bactéries. Ainsi, la division nommée « PRO » de la base EMBL (séquences des procaryotes) contient 76 215 séquences d'ARNr 16S. Récemment, l'EBI a également créé une division de la base nommée « ENV » qui contient des séquences issues d'échantillons environnementaux inconnus et dont l'annotation (notamment l'espèce) provient uniquement de l'analyse de la séquence générée et non de techniques taxonomiques classiques [Cochrane et al 2006]. Cette division de la base contient 161 645 séquences d'ARNr 16S mais on peut considérer que l'identification des organismes est moins « sûre » que pour la division PRO.

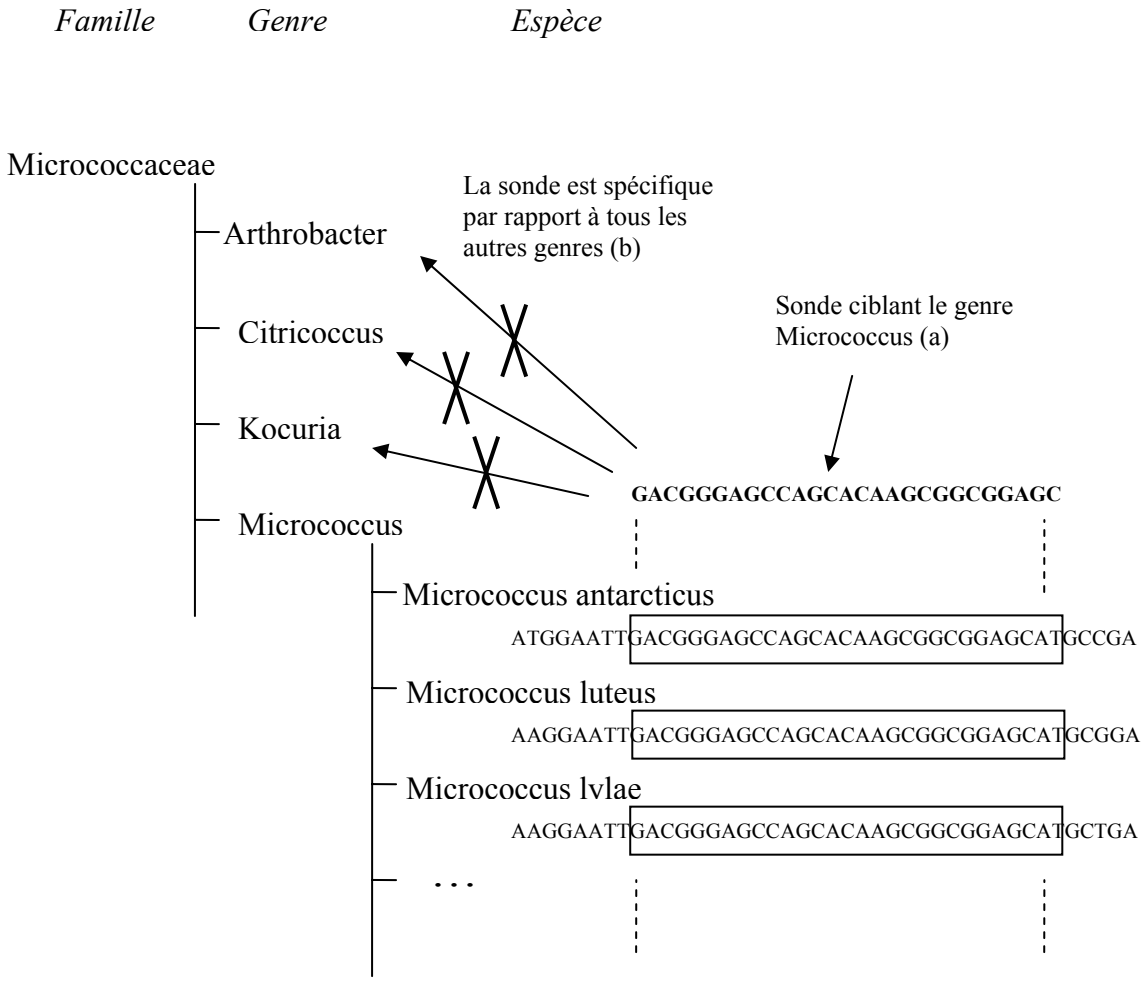


Figure 48 : La recherche de sondes phylogénétiques.

La figure illustre le mécanisme de recherche de sonde permettant d'identifier un groupe d'organismes (ici le genre *Micrococcus*). La séquence de la sonde doit être présente dans la séquence d'ARNr de l'ensemble des espèces du genre considéré (a). Elle doit également être spécifique par rapport à tous les autres organismes susceptibles de se trouver dans le mélange cible (b).

La plus importante des bases de données contenant spécifiquement des séquences d'ARNr 16S est la base RDP-II (Ribosomal Database Project II [Cole et al 2005]). Elle contient 101 632 séquences d'ARNr 16S de bactéries (release 9.21). Ces séquences sont issues des banques EMBL/GenBank/DDBJ, filtrées puis alignées contre une séquence modèle d'ARNr. Cette base apporte une annotation, un alignement des séquences ainsi qu'un placement de ces dernières dans un arbre phylogénétique.

2.3 Tests

Le problème de la détermination de sondes pour l'identification de micro organismes est extrêmement difficile pour deux raisons principales :

- Le très grand nombre de bactéries existant dans le monde vivant, ce qui se traduit par un grand nombre de séquences dans les banques de données d'ARNr 16S. Ceci multiplie le risque d'hybridation croisée lorsque l'on recherche une sonde identifiant un organisme. De plus, il existe de nombreuses erreurs d'annotation dans ces banques : certaines séquences ne sont pas attribuées à la bonne espèce, ou certains ARN 16S ont été séquencés de manière incomplète, et les différents morceaux ont étés recollés en omettant des parties. Ainsi, parmi les 101 632 séquences de la base de données RDP II release 9.21 (août 2004), seules 39 772 sont quasi complètes (longueur $\geq$ 1200 bases).
- Les séquences de la molécule d'ARNr 16S sont très similaires entre les espèces et les zones variables sont très courtes. A titre d'exemple, si l'on effectue un blast de la séquence de l'espèce *Arthrobacter globiformis* (1460 bases) contre la base RDP II, il ressort 88 séquences qui présentent une similarité de plus de 95% sur une longueur supérieure ou égale à 1360 bases, et dont le genre est différent de *Arthrobacter*. De tels genres seront pratiquement impossibles à différencier avec une seule sonde oligonucléotidique.

Le test suivant illustre le problème auquel on se trouve confronté. Imaginons que l'on souhaite déterminer une sonde spécifique du genre *Micrococcus* par rapport à l'ensemble des bactéries connues. Pour simplifier, on considère qu'il n'existe pas de variation entre les ARNr 16S des différentes espèces et on considérera donc une seule séquence (*Micrococcus luteus*) comme représentative du genre. La Figure 49 présente le résultat d'un test de spécificité de 140 sondes potentielles (50 bases) réparties sur la séquence d'ARNr 16S. Le nombre moyen d'hybridations croisées avec un autre genre pour une sonde est de 208. La sonde la plus spécifique présente 32 hybridations croisées avec d'autres genres, et donc n'est finalement absolument pas spécifique de *Micrococcus*. Ces problèmes de spécificités sont décuplés si l'on supprime les simplifications effectuées pour le test (pas de variation entre les espèces, critère de Kane simplifié…), on trouve encore plus d'hybridations croisées potentielles.

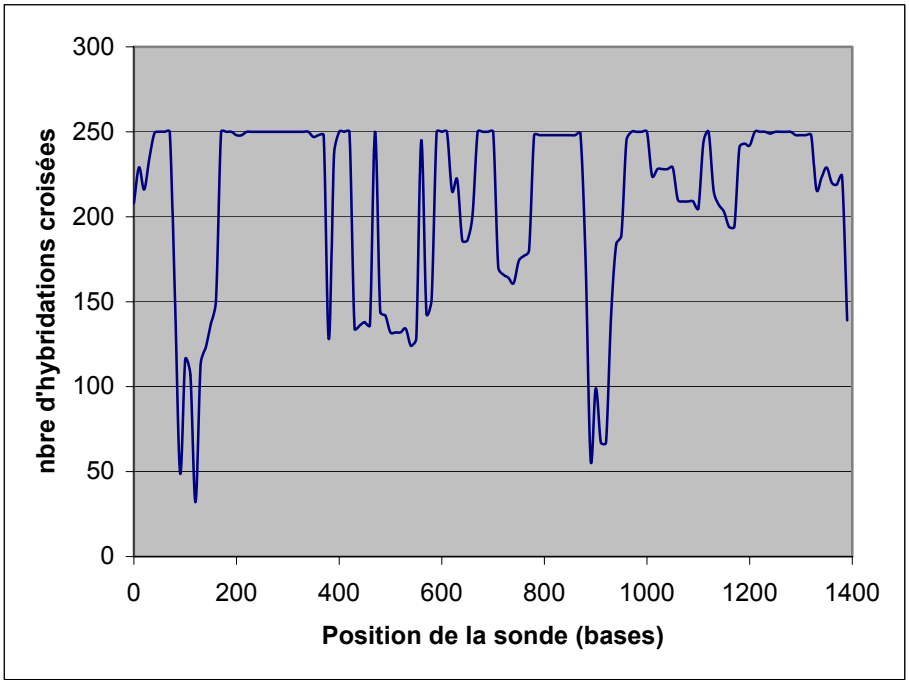


Figure 49 : Etude de la spécificité de 140 sondes ciblant le genre Micrococcus.

140 sondes (50 bases) ont été sélectionnées sur la séquence d'ARNr 16S de *Micrococcus Luteus*. Chacune de ces sondes a été blastée contre la base de données RDP II release 9.21 et le nombre d'hybridations croisées avec des séquences d'autres genres que *Micrococcus* a été comptabilisé. On considère qu'il y a hybridation croisée si la séquence présente plus de 75% de similarité avec une séquence non-cible sur la longueur de l'alignement blast (critère simplifié par rapport au critère de Kane). Le graphique présente le nombre d'hybridations croisées en fonction de la position de la sonde sur la séquence d'ARN 16S (nombre limité à 250 hybridations croisées).

Cet exemple met en évidence un problème très important : les zones de la séquence d'ARNr 16S qui différencient le plus *Micrococcus* des autres genres sont précisément les zones qui présentent le plus de variabilité entre les espèces du genre *Micrococcus*. Ainsi, on peut observer la zone où se situe la sonde la plus spécifique chez différentes espèces de *Micrococcus* :

```
AF105024   Micrococcus sp.       GGTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
AF057289   Micrococcus luteus    GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
AB023371   Micrococcus luteus    GGTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
X86609     Micrococcus sp.       GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
X86612     Micrococcus sp.       GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
X80750     Micrococcus lylae     GAGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
AF196342   Micrococcus sp.       GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
AF057290   Micrococcus lylae     GTGTGGATGAGTGGCGAACGGGTGAGTAATACGTGAGTAACCTGCCCTTG
AF196343   Micrococcus sp.       GTGTGGATGAGTGGCGAACGGGTGAGTAATACGTGAGTAACCTGCCCTTG
                                 * ***** ****************************************
```

La séquence présente des variations sur 4 nucléotides, et il n'est donc pas possible de représenter le genre *Micrococcus* par une seule séquence oligonucléotidique, si l'on choisit de sélectionner la sonde dans cette zone, qui est pourtant la plus spécifique.

2.4 Stratégie retenue

2.4.1 Pré-traitement des données

Afin de pallier aux problèmes énoncés aux paragraphes précédents, nous avons décidé tout d'abord d'effectuer un pré-traitement des bases de données d'ARNr 16S plutôt que de travailler directement sur les séquences qu'elle contiennent. Nous sommes partis de la base de données « ENV » de l'EMBL, et nous en avons extrait toutes les séquences d'ARNr 16S. Puis nous avons trié les séquences obtenues par genre et effectué les traitements suivants pour chaque genre:

- Suppression des séquences contenant plus de 10% de N (base indéterminée)
- Suppression des séquences contenant plus de dix N consécutifs
- Suppression des séquences dont la longueur est inférieure à ($L_{max,genre}$-100), où $L_{max,genre}$ est la longueur maximale des séquences du genre considéré

De plus, ces bases de données contiennent de nombreuses erreurs d'annotation : certaines séquences sont attribuées au mauvais genre. Un algorithme spécifique a été développé par Vincent Barra du LIMOS en collaboration avec le LBP afin d'éliminer le maximum de ces séquences mal annotées (en cours de publication).

L'ensemble de ce travail de filtrage a également été réalisé sur la base de données « PRO » de l'EMBL. Au final, l'utilisateur de notre programme pourra choisir d'effectuer sa recherche dans les bases ENV, PRO, ENV+PRO concaténées, ou RDP. Il pourra également utiliser sa propre banque de séquences sous forme de fichier à plat.

2.4.2 Recherche de sondes spécifiques

Etant donné que les zones de la séquence d'ARN 16S qui différencient le plus les genres entre eux se situent le plus souvent dans des régions de grande variabilité au niveau des espèces, **l'algorithme que nous avons élaboré ne va pas déterminer une sonde unique pour identifier un genre mais plusieurs sondes**. Ces sondes représenteront les variations au sein des espèces du genre ciblé. Par exemple, le résultat de recherche de sonde spécifique du genre *Micrococcus* se présentera sous la forme suivante :

G**GT**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**GG**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**AG**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**TG**TGGAT**G**AGTGGCGAACGGGTGAGTAATACGTGAGTAACCTGCCCTT**G**

Ces 4 sondes permettent d'identifier l'ensemble des espèces de *Micrococcus* présentées dans l'exemple du paragraphe 2.3. La séquence dégénérée correspondante sera également disponible :

G**DK**TGGAT**K**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**R**

(D code pour A, T ou G, K code pour G ou T, R code pour A ou G)

Ainsi, l'utilisateur souhaitant concevoir une puce à ADN phylogénétique pourra déposer les 4 sondes dans 4 spots différents pour détecter le genre *Micrococcus*. S'il le souhaite, il pourra également n'utiliser qu'un seul spot et faire synthétiser des sondes à partir de la séquence dégénérée. Mais dans ce cas, toutes les combinaisons possibles de sondes seront synthétisées, et des risques d'hybridations croisées supplémentaires seront introduits :

```
GAGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GAGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GAGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GAGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GATTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GATTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GATTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GATTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GTGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GTGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GTGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GTGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GTTTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GTTTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GTTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GTTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GGGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GGGTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GGTTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GGTTGGATGAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
GGTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
GGTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTG
```

3 Algorithme

Nous avons développé un algorithme original de conception d'oligonucléotides pour puces à ADN phylogénétiques. Cet algorithme est composé de 5 grandes étapes (détaillées dans les paragraphes suivants) :

1. Extraction des séquences du groupe d'organisme à identifier

2. Filtrage des séquences

3. Alignement multiple des séquences

4. Recherche d'une séquence consensus représentative du groupe à identifier

5. Recherche des sondes dans la séquence consensus

Les paramètres d'entrée de l'algorithme sont le groupe d'organismes à identifier G (typiquement un genre), la longueur l des oligonucléotides souhaitée, ainsi que la base de donnée utilisée B. Il est également possible de spécifier le seuil s de spécificité utilisé lors de

la recherche des hybridations croisées potentielles. En effet, on souhaite pouvoir moduler la condition (1) du critère de Kane (par défaut 75% de similarité sur la longueur de l'oligo), étant donnée la difficulté du problème si l'on conserve la valeur par défaut. Enfin, on doit spécifier le nombre maximum de bases dégénérées que peut contenir une sonde potentielle (*xdeg*) ainsi que le nombre maximum d'hybridations croisées que pourra présenter une sonde sauvegardée dans le fichier de résultat (voir plus loin).

3.1 Extraction des séquences du groupe d'organisme à identifier

La première étape consiste à extraire de la base *B* toutes les séquences du groupe *G*. Le plus souvent, les bases de données d'ARNr 16S sont des fichiers à plat au format FASTA et l'en-tête des séquences contient un identifiant unique suivant du genre et de l'espèce de l'organisme considéré. Si le groupe *G* est un genre ou une espèce, il est alors aisé de parcourir le fichier et d'en extraire les séquences qui appartiennent au groupe *G*. Dans le cas ou le groupe *G* se situe à un niveau plus élevé dans la taxonomie, nous disposons d'une base de données relationnelle contenant la taxonomie NCBI [Wheeler et al 2000] et permettant de récupérer tous les genres appartenant au groupe *G*.

3.2 Filtrage des séquences

Dans le cas où la base de donnée *B* est un fichier « à plat » fourni par l'utilisateur (qui n'a donc pas subi le pré traitement décrit au paragraphe 2.4.1), les séquences extraites sont filtrées afin de ne conserver que les séquences suffisamment longues et représentatives de l'espèce considérée.

3.3 Alignement multiple des séquences

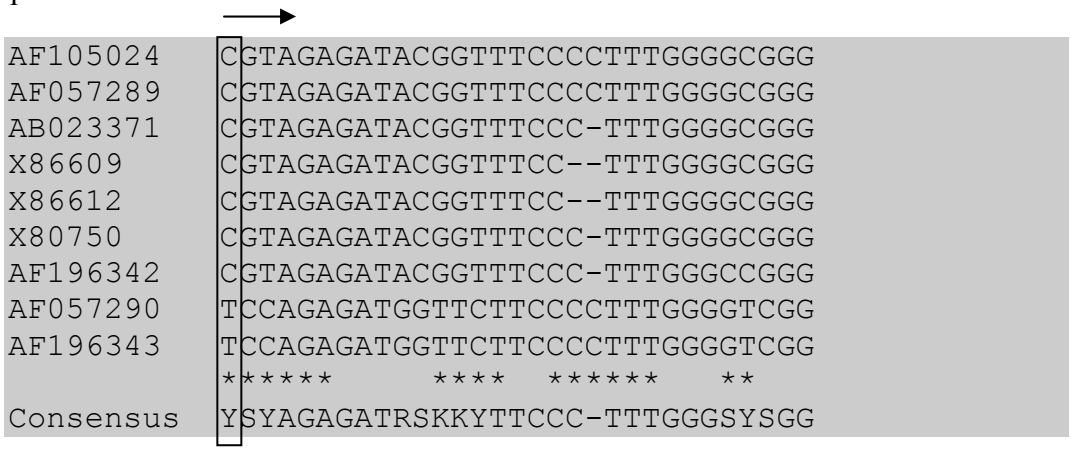
Les séquences du groupe *G* obtenues sont ensuite alignées à l'aide de l'algorithme ClustalW [Thompson et al 1994]. Ce dernier est le plus utilisé en bioinformatique pour résoudre le problème de l'alignement multiple de séquences nucléiques ou protéiques. Il s'agit d'aligner 3 séquences ou plus de façon à obtenir le plus de bases identiques en une position donnée de l'alignement. L'algorithme se déroule en trois phases :

- Calcul d'une matrice de distance entre les séquences en réalisant les alignements de toutes les séquences 2 à 2. La distance entre deux séquences est le score de l'alignement.
- Construction d'un arbre phylogénétique à partir de la matrice des distances
- Construction de l'alignement multiple en réalisant une série d'alignements 2 à 2 entre des groupes de séquences

3.4 Recherche d'une séquence consensus

L'étape suivante consiste à rechercher une séquence dite « consensus », contenant des bases dégénérées, qui soit représentative du groupe ciblé. Cette séquence est construite en parcourant l'alignement multiple colonne par colonne et en déterminant une base représentative de la colonne considérée.

Exemple :



Les règles de détermination de la « base consensus » sont les suivantes :

- Si le nombre de tiret + le nombre de N (base indéterminée) est supérieur ou égal à 50% du nombre de séquences, on ne détermine pas de base consensus, on insère un tiret dans la séquence consensus, sinon :
- La base consensus est la base dégénérée correspondant à l'ensemble des bases présentes dans la colonne sans tenir compte des tirets ou des N.

3.5 Détermination des sondes

Les sondes ciblant le groupe *G* sont recherchées dans la séquence consensus de manière similaire à la recherche de sondes spécifiques pour les puces de type transcriptome. La séquence consensus est parcourue depuis l'extrémité 3' (afin de sélectionner en priorité des sondes proches de cette extrémité) puis une sonde de longueur *l* est extraite afin d'en tester la spécificité. Seules les sondes contenant moins de *xdeg* bases dégénérées sont sélectionnées : en effet, si une sonde comporte un trop grand nombre de bases dégénérés, elle sera inutilisable concrètement puisqu'elle correspondra à un trop grand nombre de séquences réelles. Si la sonde satisfait le test de spécificité, elle est sauvegardée dans le fichier de résultat et la sonde suivante est testée en décalant la fenêtre d'une base vers l'extrémité 5' (Figure 50).

Le test de spécificité est plus complexe que dans le cas des puces de type transcriptome. En effet, étant donné une sonde contenant des bases dégénérées, il est inutile de tester la spécificité de toutes les séquences auxquelles peut correspondre la séquence consensus. Le programme ne génère donc que les séquences réelles qui étaient effectivement présentes dans au moins une des séquences d'ARNr de départ. Illustrons ce mécanisme avec les séquences

du genre *Micrococcus* présentées au paragraphe 2.4.2. Supposons que la séquence de la sonde à tester soit :

G**DK**TGGAT**K**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**R**

L'algorithme va rechercher les séquences qui ont servi à générer la séquence consensus et uniquement celles-ci :

G**GT**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**GG**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**AG**TGGAT**T**AGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTT**A**
G**TG**TGGAT**G**AGTGGCGAACGGGTGAGTAATACGTGAGTAACCTGCCCTT**G**

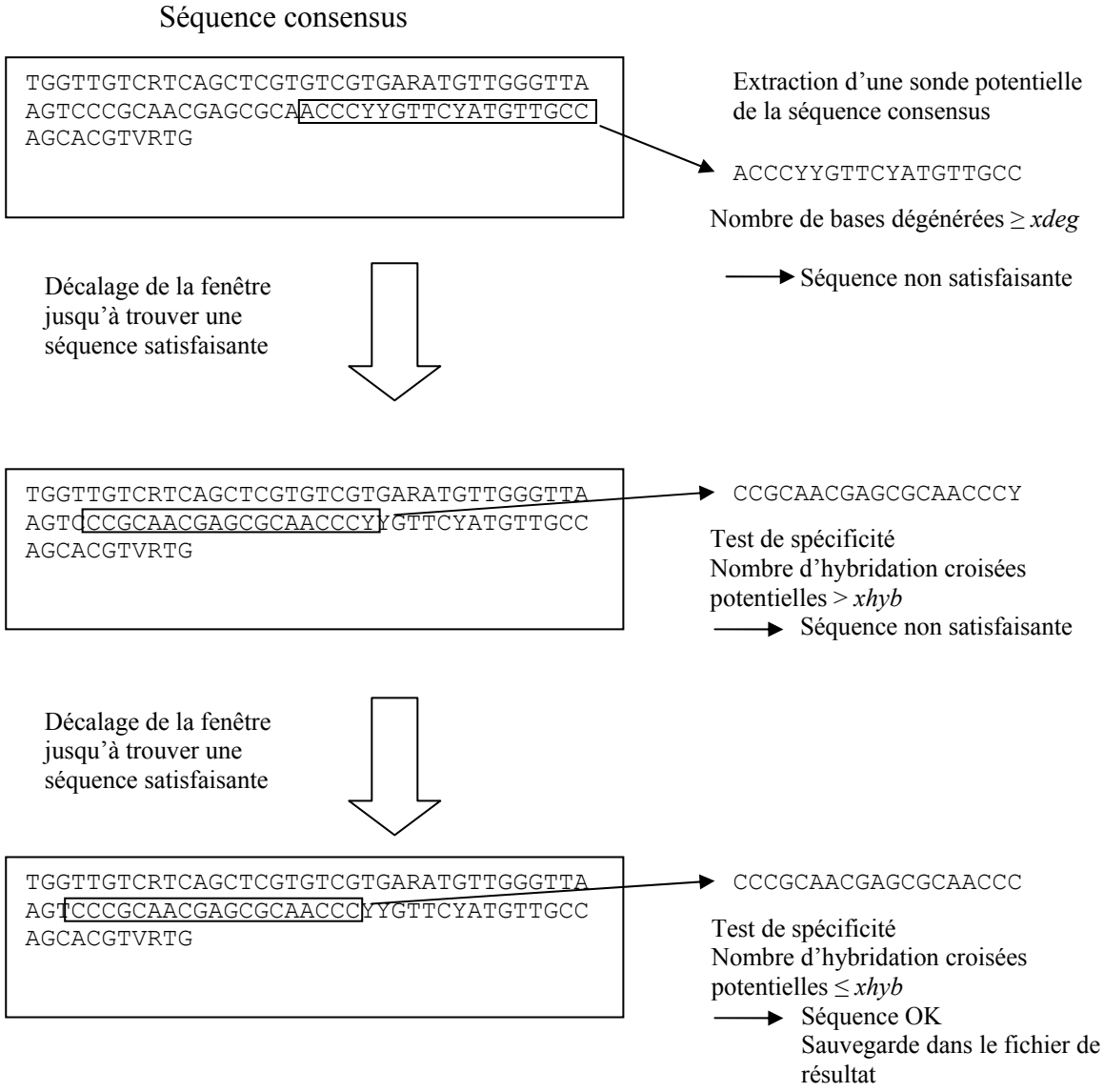


Figure 50 : Schéma général de la détermination des sondes à partir de la séquence consensus.

Les paramètres utilisés sont l=19, xdeg=1.

Le programme effectue alors un blast de chacune de ces séquences contre la base de données *B*, afin de tester si elles satisfont le critère de Kane. En cas de réponse négative, la séquence n'est pas rejetée, mais le nombre d'hybridations croisées potentielles est mémorisé ainsi que les genres correspondant à ces hybridations croisées. Si le nombre total d'hybridations croisées est inférieur ou égal à *xhyb*, la sonde est alors considérée comme satisfaisante et toutes les informations sont sauvegardées dans le fichier de résultat (Figure 51).

```
Sonde consensus :
GDKTGGATKAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTR
Position : 1034
Oligonucleotides reels :
GGTTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
Croise avec :

GGGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
Croise avec :

GAGTGGATTAGTGGCGAACGGGTGAGTAACACGTGAGTAACCTGCCCTTA
Croise avec :

GTGTGGATGAGTGGCGAACGGGTGAGTAATACGTGAGTAACCTGCCCTTG
Croise avec :
```

Figure 51 : Contenu du fichier de résultat (pour une sonde donnée).

La Figure 52 résume l'ensemble des étapes de l'algorithme de recherche de sondes.

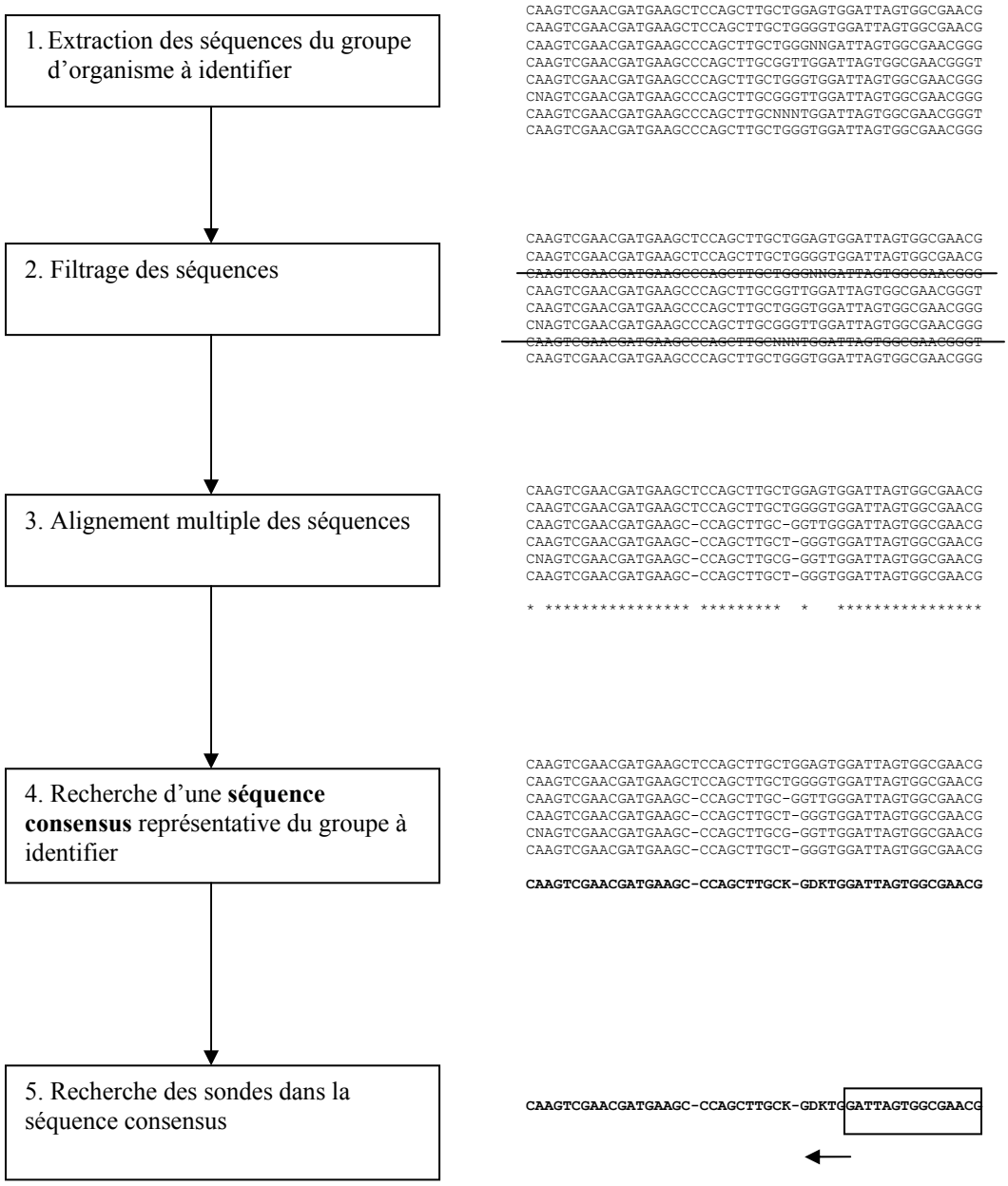


Figure 52 : Représentation schématique de l'algorithme de recherche de sondes pour puces phylogénétiques.

4 Le logiciel PhylArray

4.1 Architecture

PhylArray est un logiciel écrit en langage Perl qui implémente notre algorithme de recherche de sondes pour puces à ADN phylogénétiques. Il utilise des bases de données relationnelles pour stocker les séquences d'ARNr ainsi que la taxonomie NCBI. Ces bases ont été implémentées avec le Système de Gestion de Bases de Données MySql. La Figure 53 présente l'architecture du logiciel.

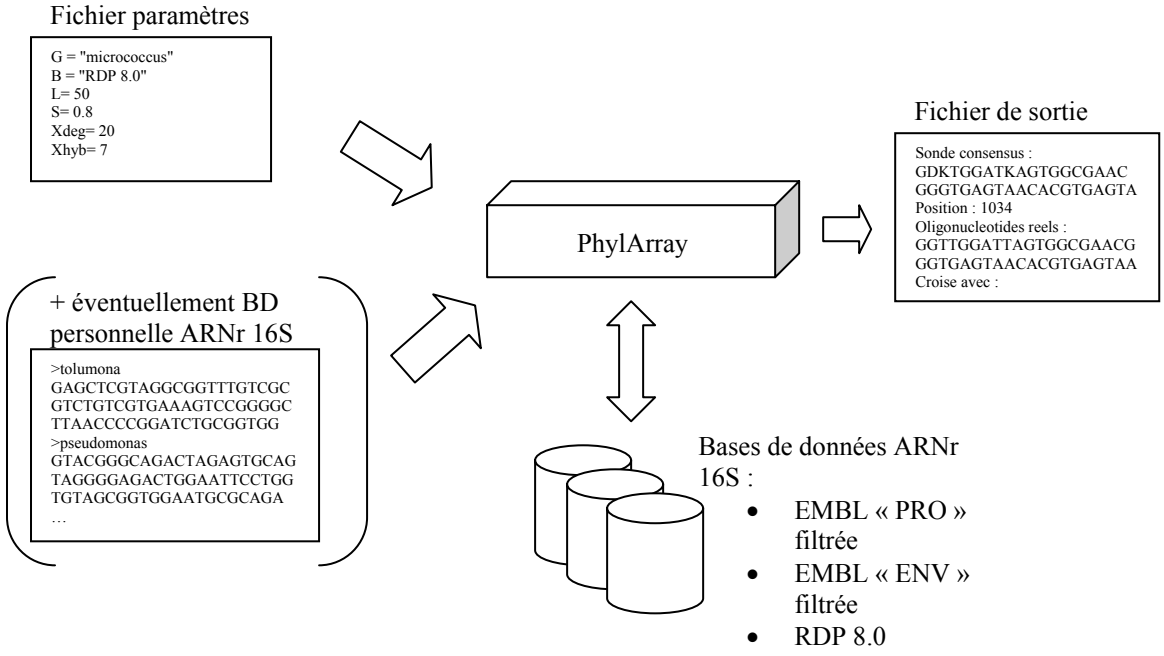


Figure 53 : Le logiciel PhylArray - Schéma de flot de données.

4.2 Exemple d'exécution

Nous présentons ici le détail de l'exécution de PhylArray avec les paramètres d'entrée suivants :

```
G     = "micrococcus"
B     = "RDP 8.0"
L     = 50
S     = 0.8
Xdeg  = 20
Xhyb  = 7
```

Ce qui signifie que l'on recherche des sondes de 50 nucléotides ciblant le genre *Micrococcus*. Ces sondes devront être les plus spécifiques possibles par rapport à tous les autres genres présents dans la base de données RDP 8.0. Le seuil de spécificité utilisé dans la condition (1)

du critère de Kane sera de 80%. Les sondes « dégénérées » ainsi déterminées ne devront pas comporter plus de 20 bases dégénérées et les sondes réelles correspondantes ne devront pas croiser avec plus de 7 genres au total.

1. Extraction des séquences du genre *Micrococcus*

Les séquences appartenant au genre *Micrococcus* sont extraites de la base de données RDP 8.0. On obtient 11 séquences ayant pour en tête FASTA :

```
>AB023371    Micrococcus luteus
>AF057289    Micrococcus luteus
>M38242      Micrococcus luteus
>AF057290    Micrococcus lylae
>X80750      Micrococcus lylae
>AF105024    Micrococcus sp. DD75
>AF196342    Micrococcus sp. SMCC ZAT351
>AF196343    Micrococcus sp. SMCC ZAT352
>X86608      Micrococcus sp.
>X86609      Micrococcus sp.
>X86612      Micrococcus sp.
```

2. Filtrage des séquences

La base de données utilisée est associée au logiciel PhylArray. Ainsi, les séquences de celle-ci étaient déjà prétraitées, et donc cette étape de filtrage ne supprime aucune séquence.

3. Alignement multiple des séquences

Le programme ClustalW est exécuté sur les séquences extraites. Voici un extrait de l'alignement obtenu :

```
AF057290        CAAGTCGAACGATGAAGCTCCAGCTTGCTGGAGTGGATTAGTGGCGAACGGGTGAGTAAC
X80750          CAAGTCGAACGATGAAGCTCCAGCTTGCTGGGGTGGATTAGTGGCGAACGGGTGAGTAAC
M38242          CAAGTCGAACGATGAAGC-CCAGCTTGCT-GGGNNGATTAGTGGCGAACGGGTGAGTAAC
X86608          CAAGTCGAACGATGAAGC-CCAGCTTGC--GGTTGGATTAGTGGCGAACGGGTGAGTAAC
AB023371        CAAGTCGAACGATGAAGC-CCAGCTTGCT-GGGTGGATTAGTGGCGAACGGGTGAGTAAC
X86609          CNAGTCGAACGATGAAGC-CCAGCTTGCG-GGTTGGATTAGTGGCGAACGGGTGAGTAAC
X86612          CAAGTCGAACGATGAAGC-CCAGCTTGC--NNNTGGATTAGTGGCGAACGGGTGAGTAAC
AF057289        CAAGTCGAACGATGAAGC-CCAGCTTGCT-GGGTGGATTAGTGGCGAACGGGTGAGTAAC
AF105024        CAAGTCGAACGATGAAGC-CCAGCTTGCT-GGGTGGATTAGTGGCGAACGGGTGAGTAAC
AF196342        CAAGTCGAACGCTGAAGCACCAGCTTGCTGGTGTGGATGAGTGGCGAACGGGTGAGTAAT
AF196343        CAAGTCGAACGCTGAAGCACCAGCTTGNTGGTGTGGATGAGTGGCGAACGGGTGAGTAAT
                * ********* ****** ********        *** ********************
```

4. Recherche d'une séquence consensus représentative du groupe à identifier

La séquence consensus obtenue est :

```
--------------TTTGATCMTGGCTCAGGAYGAACGCTGGCGGCGTGCTTAACACATGCAAGTCGAACGMTGAAGC-CCA
GCTTGCK-GDKTGGATKAGTGGCGAACGGGTGAGTAAYACGTGAGTAACCTGCCCTTRACTCTGGGATAAGCCYKGGAAACK
RGGTCTAATACYGGATRRKASMDBYYAYCGCATGGTGG-GTGTKGRAARGRWTTAYYGGTYWTGGATGGRCTCRCGGCCYAT
CAGCTTGTTGGTGAGGTAATGGCTCACCAAGGCGACGACGGGTAGCCGGCCTGAGAGGGTGACCGGCCACACTGGGACTGAG
ACACGGCCCAGACTCCTACGGGAGGCAGCAGTGGGGAATATTGCACAATGGGCGMAAGCCTGATGCAGCGACGCCGCGTGAG
GGATGACGGSCTTCGGGTTGTAAACCTCTTTCAGYAGGGAAGAAGCSA-AAGTGACGGTACCTGCAGAAGAAGCRCCGGCTA
ACTACGTGCCAGCAGCCGCGGTAATACGTAGGGYGCRAGCGTTRTCCGGAATTATTGGGCGTAAAGAGCTCGTAGGCGGTTW
GTYGCGTCTGYYGTGAAAGYCCGGGGCTTAACYCCGGRTSTGCRGTGGGTACG-GGCAGACT-AWGAGTGCAGTAGGGGAGA
CTS-GAATTCCT-GGTGTAGCGGTGGAATGCGCAGATATCAGGAGGAACACCGATGGCGAAGGCAGGTCTCTGGGCTGTWAC
TGACGCTGAGGAGCGAAAGCATGGGGAGCGAACAGGATTAGATACCCTGGTAGTCCATGCCGTAAACGTTKGSSACTASGTG
TGGGGRMCATTCCACGKTTTCCGCGCCGYASSTAACGCATTAAGTGCCCCGCCTGGGGAGTACGGCCGCAAGGCTAAAACTC
```

```
AAAGGAATTGACGGGGGCCSGCACAAGCGGCGGAGCATKCGKATTAATTCGATGCAACGCGAAGAACCTTACCAAGGCTTGA
CATRTWCYSGWTCGYYSYAGAGATRSKKYTTCCCCTTTGGGSYSGGTWBACAGGTGGTGCA-TGGTTGTCRTCAGCTCGTGT
CGTGARATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCYYGTTCYATGTTGCCAGCACGTVRTGGTGGGGACTCATGGRAG
ACTGCCGGGGTCAACTCGGAGGAAGGTGDRGAYGACGTCAAATCATCATGCCCCTTATGTCTTGGKCTTSASSSATGCTAMA
ATGGCCGGTACAAWGGGTTGCRATACTGTRAGGTGGAGCTAATCCCAAAARGCCGGTCTCAGTTCGGATTGRGGTCTGCAAC
TCGACCYCATGAAGTYGGAGTC-GCTAGTAATCGCAGATCMGCMACGCKGCGGTGAATACGTTC-CGGGCCTTGTACACACC
GCCCGTCAAGTCACGAAAGTYGGTAACACCCGAA-CCGGTGGCSTAA----------------------------------
------------------------------------------------------------------
```

5. Recherche des sondes dans la séquence consensus

Voici deux exemples de sondes trouvées par le programme :

```
Sonde consensus :
TCGYYSYAGAGATRSKKYTTCCCCTTTGGGSYSGGTWBACAGGTGGTGCA
Position : 546

Oligonucleotides reels :
TCGCCGTAGAGATACGGTTTCCCCTTTGGGGCGGGTTCACAGGTGGTGCA
Croise avec :
Bifidobacterium
Corynebacterium
Rhodococcus

TCGCCGTAGAGATACGGTTTCCCCTTTGGGCCGGGTACACAGGTGGTGCA
Croise avec :
Bifidobacterium
Corynebacterium
Rhodococcus

TCGTTCCAGAGATGGTTCTTCCCCTTTGGGGTCGGTATACAGGTGGTGCA
Croise avec :
Bifidobacterium
Corynebacterium
Rhodococcus
Kocuria
Terrabacter
Janibacter
Streptomyces

TCGCCGTAGAGATACGGTTTCCCCTTTGGGGCGGGTTGACAGGTGGTGCA
Croise avec :
Rhodococcus
Kocuria
Terrabacter
Janibacter

TCGCCGTAGAGATACGGTTTCCCCTTTGGGCCGGGTTCACAGGTGGTGCA
Croise avec :
Bifidobacterium
Corynebacterium
Rhodococcus

Sonde consensus :
TTGACATRTWCYSGWTCGYYSYAGAGATRSKKYTTCCCCTTTGGGSYSGG
Position : 561

Oligonucleotides reels :
TTGACATGTTCTCGATCGCCGTAGAGATACGGTTTCCCCTTTGGGGCGGG
Croise avec :
Arthrobacter
```

```
TTGACATGTTCTCGATCGCCGTAGAGATACGGTTTCCCCTTTGGGCCGGG
Croise avec :
Arthrobacter

TTGACATGTTCTCGTTCGCCGTAGAGATACGGTTTCCCCTTTGGGGCGGG
Croise avec :
Arthrobacter

TTGACATATACCGGATCGTTCCAGAGATGGTTCTTCCCCTTTGGGGTCGG
Croise avec :
Arthrobacter
Kocuria
```

4.3 Utilisation du logiciel

Le logiciel PhylArray a été utilisé par les biologistes du LBP pour concevoir des premières puces « tests », afin de vérifier expérimentalement la validité des sondes calculées. Les premiers résultats (non détaillés ici) sont encourageants, et il semble que les sondes calculées à l'aide de PhylArray aient une meilleure spécificité que celle déterminées par d'autres logiciels du même type (PRIMROSE, ARB).

Nous avons également souhaité travailler sur l'efficacité de l'algorithme et il nous est apparu que celui-ci serait aisément parallélisable. En effet, le calcul d'une sonde spécifique d'un genre donné comporte de nombreuses opérations totalement indépendantes, notamment au cours de l'alignement multiple et des tests de spécificité.

5 Parallélisation de l'algorithme

5.1 Principe

Les deux phases de l'algorithme les plus coûteuses en temps machine sont l'alignement multiple des séquences à l'aide de l'algorithme ClustalW et la recherche de sondes dans la séquence consensus (test de spécificité). En comparaison, les autres étapes ont des temps d'exécution négligeables. Or ces deux étapes sont aisément parallélisables car, comme dans beaucoup d'algorithmes en bioinformatique, il est possible de découper les données en « paquets » pour ensuite réaliser les actions sur ces données de manière totalement indépendante (parallélisation sur les données).

De nombreux travaux ont été effectués sur la parallélisation de l'algorithme Clustalw. Ebedes and Datta [2004] remarquent que lors d'une exécution de l'algorithme, 96% du temps est passé sur la première phase, c'est à dire l'alignement de toutes les séquences deux à deux. Le gain de temps se fait donc principalement en parallélisant cette étape de l'algorithme. Pour *n* séquences, *n(n-1)/2* alignements doivent être calculés, tous ces calculs étant totalement indépendants. La difficulté réside alors uniquement dans l'équilibrage de charge entre les différentes machines (ou processeurs, suivant l'architecture parallèle sur laquelle on travaille). Mikhailov et al [2001] proposent une implémentation parallèle de l'algorithme Clustalw pour machines multiprocesseurs à mémoire partagée, tandis que Ebedes and Datta [2004] présentent une implémentation pour les architectures de type clusters[20]. Pour paralléliser l'étape d'alignement multiple de PhylArray, nous avons utilisé l'implémentation de Li [2003] qui est une parallélisation de ClustalW pour les architectures de type cluster utilisant la libraire de passage de messages MPI (Message Passing Interface).

La phase de recherche de sondes spécifiques dans la séquence consensus est également facilement parallélisable. En effet, chaque sous-séquence de longueur *l* de la séquence consensus est testée et tous les tests sont complètement indépendants. Si l'on souhaite découper cette étape en *p* calculs indépendants, il suffit de découper la séquence consensus en *p* parties. Schématiquement, l'algorithme est transformé en le paramétrant avec l'indice de début et de fin de recherche à l'intérieur de la séquence consensus (les algorithmes sont donnés en langage algorithmique formel) :

Algorithme initial :

```
lg= longueur(séquence consensus)
Pour i de lg-ls+1 à 1 [pas -1]
| Sonde = séquence consensus[i, i+ls]
| Tester (Sonde)
FinPOur
```

où seq[i, j] désigne la sous-séquence de seq de l'indice i à l'indice j, et ls la longueur des sondes recherchées (notée précedemment *l*)

Algorithme parallélisé :

```
Fonction recherche(debut, fin)
| lg= longueur(séquence consensus)
|      Pour i de fin-ls à debut
|      | Sonde = séquence consensus[i, i+ls]
|      | Tester (Sonde)
| FinPour


lg= longueur(séquence consensus)
Pour i de 0 à p-1
| d=i*lg/p
| f=(i+1)*lg/p) -1
| recherche(d,f)
FinPour
```

où p est le nombre de processus que l'on souhaite générer

[20] Ou « ferme de calcul » : groupe d'ordinateurs en réseau fonctionnant comme un seul et même système afin d'augmenter la capacité de calcul et de stockage.

5.2 Implémentation sur une architecture de type cluster

Le logiciel PhylArray ainsi modifié a été déployé sur une architecture de type cluster de calcul. Le cluster se compose de 15 machines : une machine maître (noeud de management) et 14 noeuds (de calcul). Ces noeuds sont des machines bi-processeurs Xeon hyperthreadés à 2.67 GHz, ayant chacune 2 Go de RAM et munie d'un disque 73 Go SCSI.

Les phases 1, 2 et 4 de l'algorithme s'exécutent sur le nœud maître.

Sur ce cluster ont été installés la bibliothèque MPI ainsi que le logiciel ClustalW-MPI, afin de paralléliser la phase 3 de l'algorithme PhylArray. L'utilisation est similaire au programme ClustalW classique et le lancement s'effectue sur le nœud maître.

La parallélisation de la phase 5 de PhylArray utilise le gestionnaire de batch OpenPBS. OpenPBS est une version libre de Portable Batch System, un gestionnaire de travaux « batch » développé pour la NASA au début des années 90. Il est utilisé dans les environnements de type cluster pour répartir les tâches (appelées « jobs » dans ce contexte) soumises par les utilisateurs sur les différents nœuds de calculs. Pour exécuter un calcul sur le cluster, l'utilisateur fournit un script ayant l'extension .pbs au gestionnaire de jobs sur le nœud maître. Concrètement, un script .pbs est un programme classique qui est exécuté par l'interpréteur de commande (script shell). Ce script contient certaines directives à destination du gestionnaire de job (durée maximale du job, nombre de nœuds nécessaires). Lorsque l'on soumet un job au gestionnaire, celui-ci est placé dans une file d'attente (la gestion des jobs est purement FIFO). Lorsque vient le tour du job, openPBS recherche une ou plusieurs machines susceptibles d'accueillir le job, en mesurant la charge de celles-ci. L'exécution du script est ensuite lancée sur ces machines.

6 Réalisation d'un portail Web pour le lancement de PhylArray

Les chercheurs de l'équipe Génomique Intégrée des Interactions Microbiennes, répartis sur différents sites, souhaitaient pouvoir exécuter le logiciel PhylArray à distance. Afin de répondre à ce besoin, un portail Web a été réalisé et mis à disposition de l'ensemble de la communauté bioinformatique. Il permet à tout utilisateur préalablement enregistré d'exécuter le logiciel sur le cluster en accédant à l'interface via un navigateur Web.

6.1 Présentation de l'interface

L'interface Web de PhylArray est accessible à l'adresse suivante :
http://fc.isima.fr/~rimour/phylarray/

La page d'accueil permet aux utilisateurs enregistrés de se connecter (Figure 54). Le menu de gauche permet d'accéder à une page d'information ainsi qu'à une page d'enregistrement des nouveaux utilisateurs.

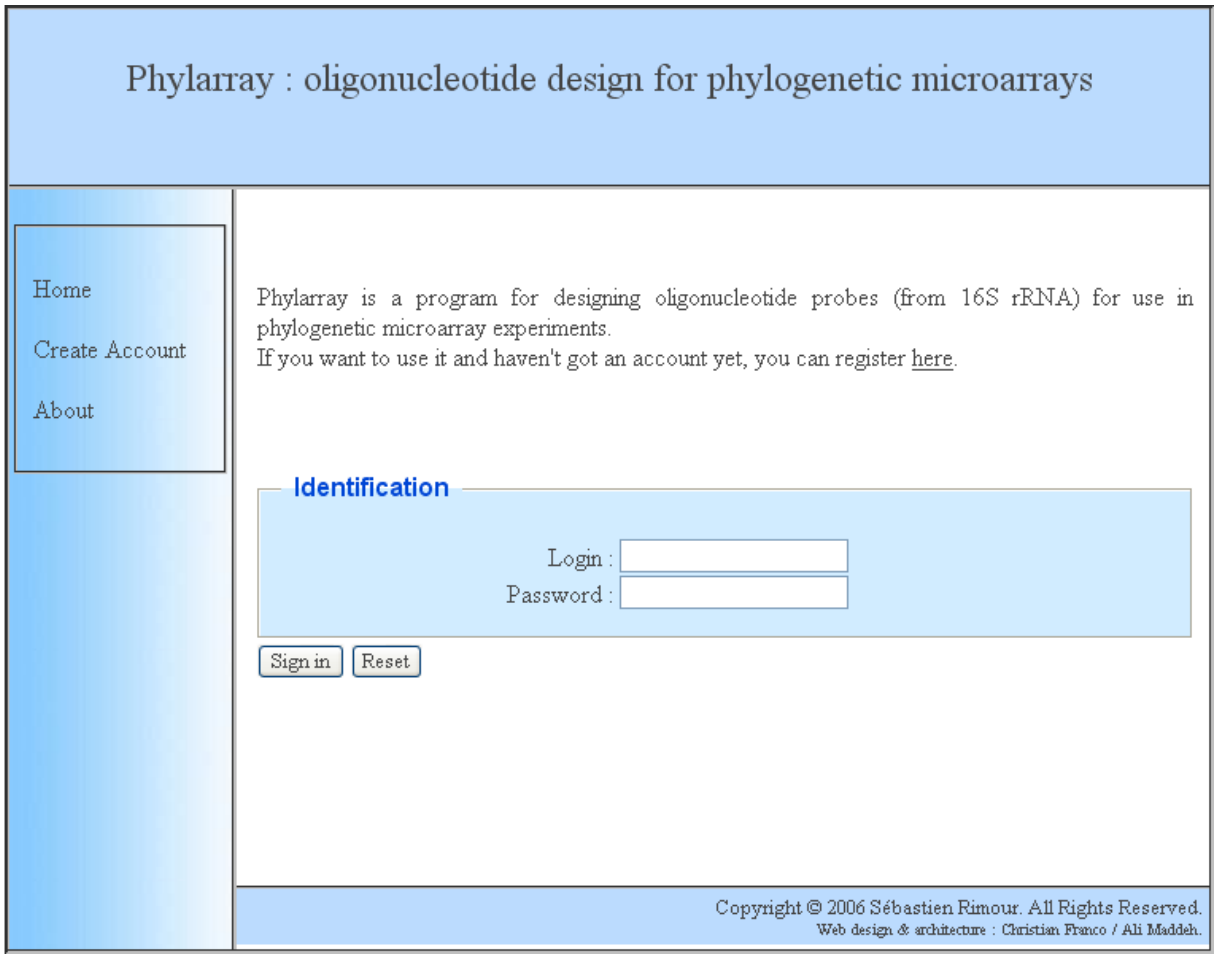


Figure 54 : La page d'accueil de PhylArray.

Une fois connecté l'utilisateur dispose de trois possibilités via le menu de gauche : lancer un nouveau calcul (« New job »), visualiser l'état des calculs en cours (« Running jobs »), et voir les résultats des calculs terminés (« Old jobs »).

- Lancement d'un nouveau calcul :

L'utilisateur entre les différents paramètres nécessaires à PhylArray via un formulaire puis lance le calcul (Figure 55). Ce dernier est effectué sur le cluster comme décrit au paragraphe 5.2. Dans le cas où un calcul a déjà été effectué avec les mêmes paramètres d'entrées et si le fichier de résultat est toujours disponible dans la partie « Old jobs », le calcul n'est pas lancé et l'utilisateur est renvoyé vers le fichier de résultat.

- Visualisation des calculs en cours :

Cette option permet de visualiser quels sont les calculs en cours d'exécution sur le cluster ainsi que leurs différents paramètres.

- Récupération des résultats :

Lorsqu'un calcul est terminé, un courrier électronique est envoyé à l'utilisateur l'invitant à se connecter à l'interface PhylArray. Le fichier de résultat est alors téléchargeable dans la rubrique « Old jobs » (Figure 55). Les différents fichiers de résultats sont conservés sur le serveur tant que l'utilisateur ne les supprime pas.

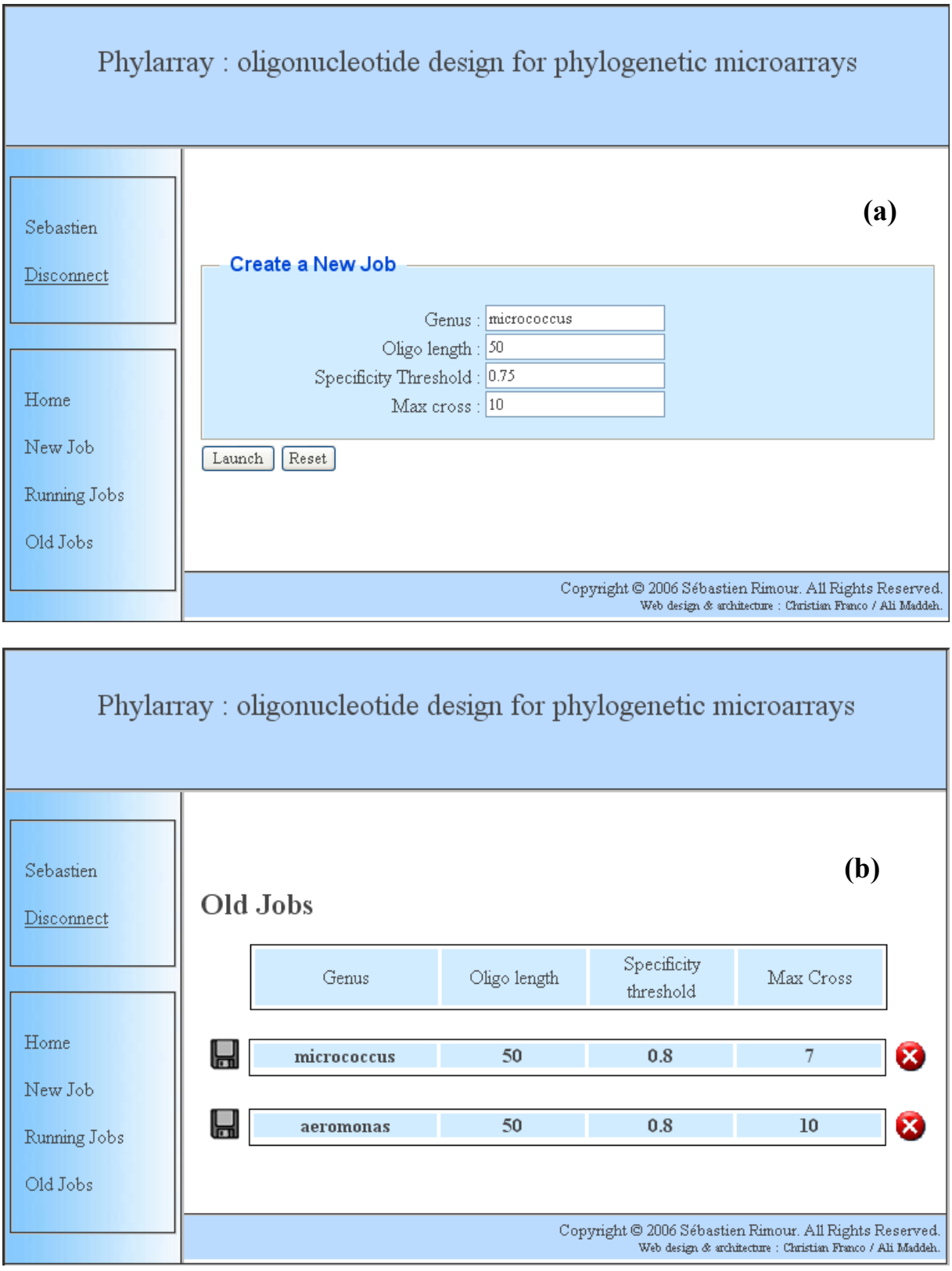


Figure 55 : Les pages de lancement d'un nouveau calcul (a) et de visualisation des calculs terminés (b).

6.2 Architecture de l'application

L'architecture de l'application est présentée Figure 56 . La partie interface est constituée d'un serveur Web (Apache), et le site est constitué de pages XHTML et PHP. Cette partie a été réalisée par deux étudiants de l'ISIMA (Institut Supérieur d'Informatique de Modélisation et de leurs Applications) au cours d'un projet que j'ai encadré [Franco et Maddeh 2006]. Des scripts Perl assurent la communication avec le nœud maître du cluster sur lequel s'exécute l'application PhylArray. L'ensemble des informations relatives aux utilisateurs et aux calculs lancés est conservé dans une base de données MySQL.

La Figure 57 présente le diagramme des cas d'utilisation de l'interface Web. Les cas d'utilisation principaux pour la gestion des utilisateurs sont « l'inscription » et « l'authentification ». Une inscription effectuée par un utilisateur entraîne des opérations du côté du serveur Web comme l'enregistrement de l'utilisateur de manière persistante et un envoi de mail de notification d'inscription. L'authentification va permettre à un chercheur d'accéder aux services de gestion de jobs.

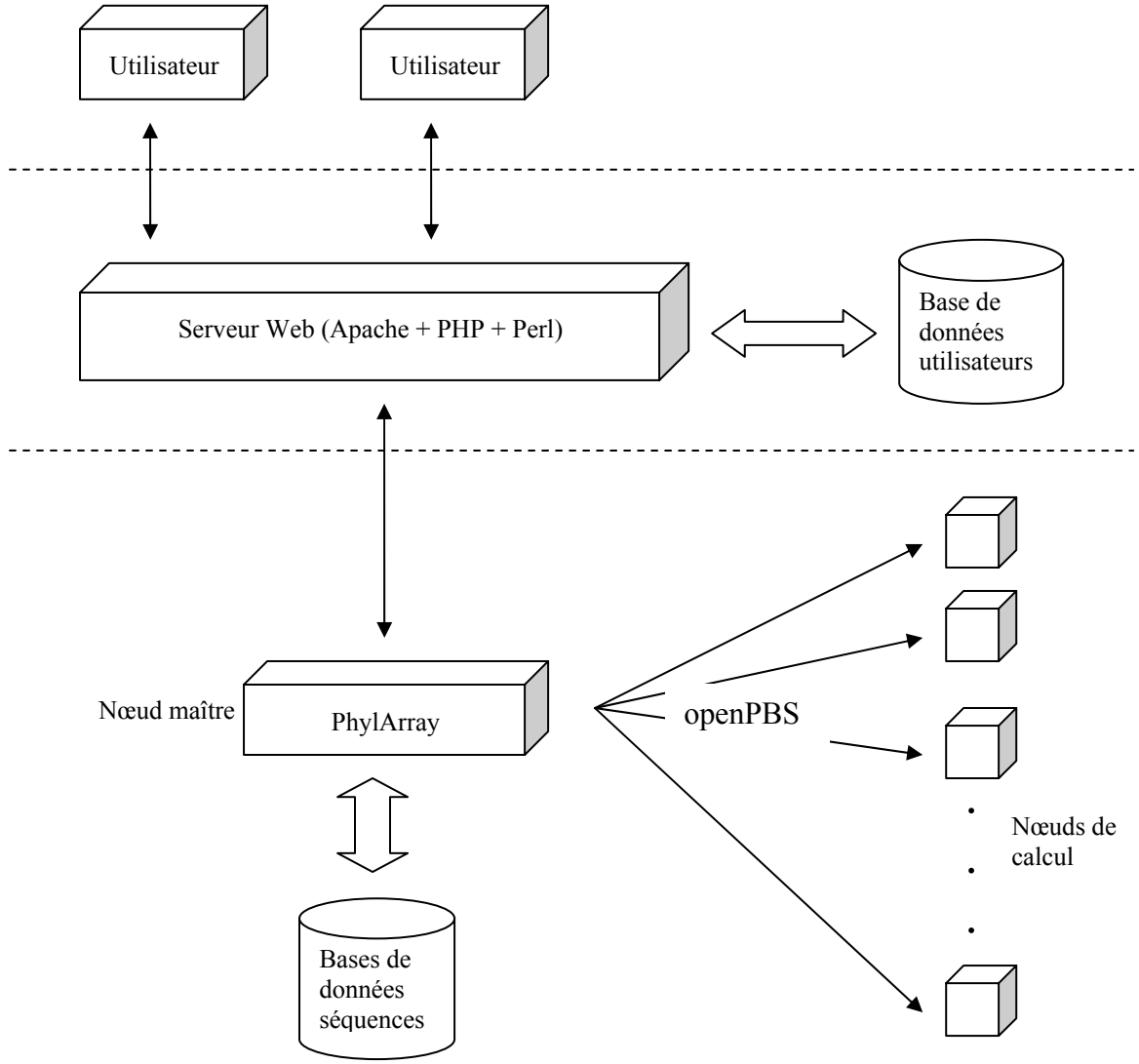


Figure 56 : Architecture de l'application PhylArray.

Les besoins les plus importants sont le « lancement d'un job » et la « gestion des anciens jobs ». Le rôle du serveur Web comprendra l'enregistrement des jobs effectués et le lancement de « PhylArray ». Il participera à l'authentification sous la forme d'autorité de validation.

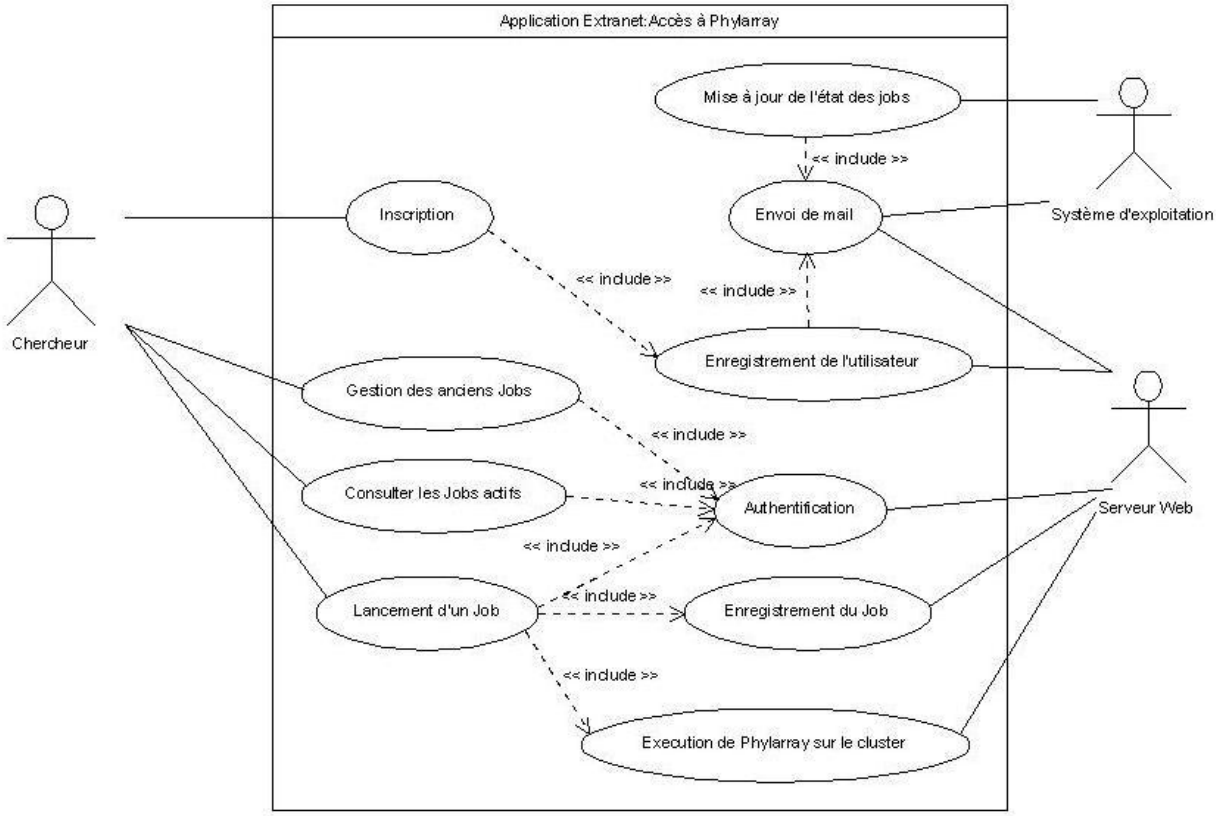


Figure 57 : Diagramme des cas d'utilisation de l'interface Web.

L'ensemble de ces cas d'utilisation a été implémenté et le tout fonctionne conformément au cahier des charges. Le mécanisme des « sessions » PHP a été utilisé pour permettre à l'utilisateur de naviguer sur toutes les pages en ne s'identifiant qu'une seule fois. Une session est un fichier conservé sur le serveur et accessible aux scripts en fonction d'un identifiant généré à la création. Chaque fois qu'un visiteur s'authentifie, il va générer une session où un identifiant lui sera attribué. Nous avons choisi les sessions et non les cookies car elles sont plus sécurisées car stockées sur le serveur et non chez le client comme pour les cookies.

A titre d'exemple, la Figure 58 présente le cas d'utilisation central de l'application : le lancement d'un job par l'utilisateur. Le scénario nominal se déroule comme suit : à la réception du formulaire, les données sont lues et formatées pour être communiquées au logiciel PhylArray, ensuite quelques manipulations de fichiers sont nécessaires pour préparer le lancement de PhylArray, enfin une connexion est réalisé au cluster par le serveur Web aboutissant au lancement du calcul. Des enchaînements alternatifs peuvent être effectués pour les raisons suivantes :

- Comme pour l'inscription, tous les champs doivent être saisis dans le formulaire, en effet le logiciel PhylArray ne fonctionnera pas si tous les paramètres ne sont pas renseignés.

- De plus, il est demandé que chaque utilisateur ne puisse pour le moment lancer qu'un seul job à la fois sur le serveur, afin de ne pas le surcharger.
- Enfin, étant donné que notre application doit gérer les anciens jobs en garantissant l'accès à leur résultat dans le temps, une des exigences est de ne pas autoriser un utilisateur à lancer un job avec des paramètres similaires à un job dont il possède déjà le résultat.

Dans les trois cas définis précédemment, des messages clairs indiquent les démarches à suivre pour utiliser à bien le service en cas de problème.

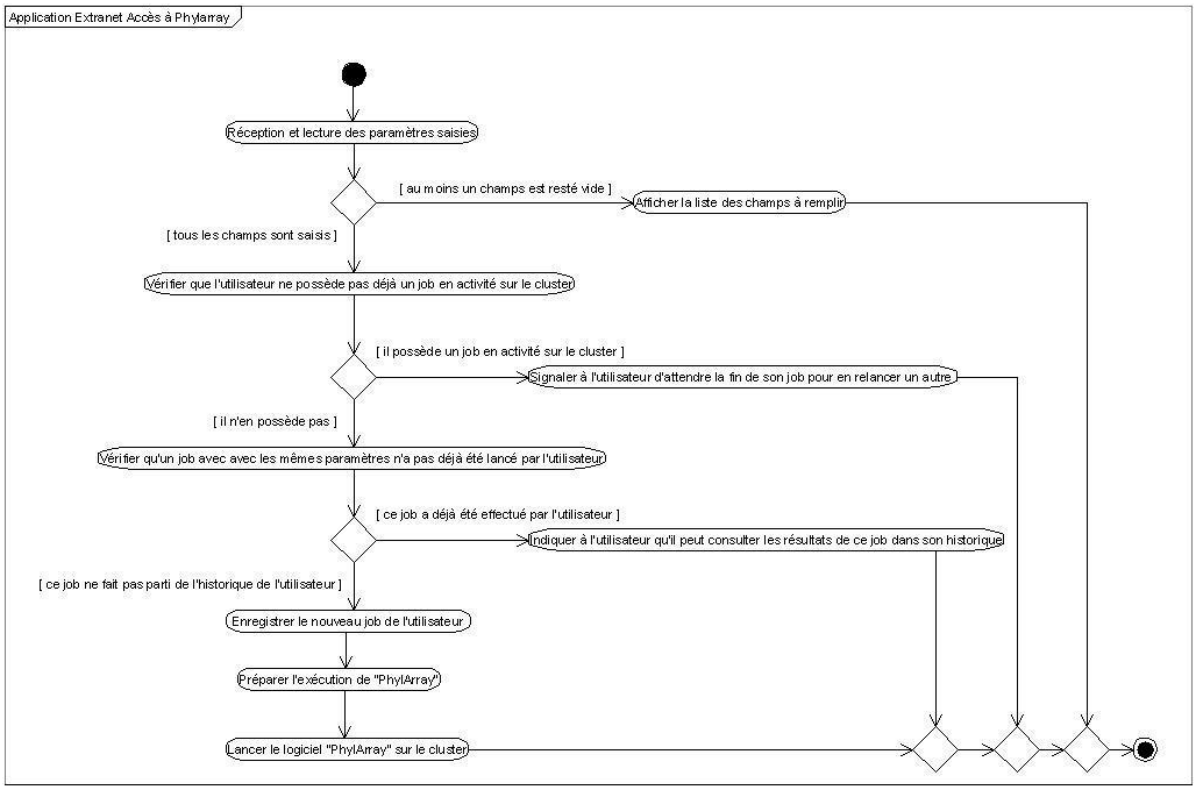


Figure 58 : Diagramme d'activité correspondant au lancement d'un job.

7 Conclusion

Dans ce chapitre, nous avons appliqué les connaissances acquises dans le domaine de la conception des sondes pour les biopuces transcriptomiques à un domaine voisin : celui-ci des puces phylogénétiques. En effet, les biologistes du LBP avec qui nous collaborons souhaitaient également réaliser de telles puces. Leur but est de suivre l'évolution des

principales communautés bactériennes au cours d'un processus de bioremédiation. Nous avons vu que les logiciels permettant de concevoir des sondes dans ce cadre sont peu nombreux.

Nous avons proposé un algorithme original de détermination d'oligonucléotides ciblant l'ARNr 16S. Cet algorithme a été implémenté dans le logiciel PhylArray, puis parallélisé afin de réduire les temps de calcul. Les résultats des premières expériences de biopuces conçues à l'aide de PhylArray sont encourageants. Néanmoins, plusieurs améliorations restent à apporter à l'interface Web du logiciel. L'ergonomie pourra notamment être amélioré en permettant à l'utilisateur de parcourir la taxonomie afin de sélectionner les groupes d'organismes à cibler.

Enfin, une des perspectives majeures est le couplage de l'algorithme de PhylArray avec notre nouvelle approche de conception de sondes utilisant des sondes composites. Un tel couplage pourrait permettre d'augmenter encore la spécificité des sondes obtenues.

Conclusion et perspectives

Synoptique de nos travaux

La « bioinformatique des puces à ADN » est un domaine extrêmement vaste puisqu'il englobe des disciplines aussi différentes que l'algorithmique, le traitement d'image, les bases de données, la statistique, et la fouille de données. L'utilisation de cette technologie est en pleine expansion, et la communauté de chercheurs travaillant dans ce domaine est très active, à tel point qu'il est relativement difficile de recenser l'ensemble des méthodes et logiciels susceptibles de répondre à un problème donné.

Après m'être intéressé à l'ensemble des étapes informatiques clés d'une expérience de biopuce, j'ai choisi de concentrer mes travaux de recherche sur l'étape de conception de l'expérience, c'est-à-dire la détermination des séquences des sondes qui seront déposées sur le support solide. En effet, cette question était au centre des préoccupations de l'équipe de biologistes avec qui je collaborais. Cette étape est très importante car la pertinence des résultats obtenus à l'issu de l'expérience dépendra fortement de la capacité des sondes à « reconnaître » leur cible et uniquement celle-ci. L'objectif est de fournir de nouvelles méthodes et de nouveaux algorithmes afin notamment d'améliorer les résultats obtenus par les outils existants. En effet, ces derniers ne donnaient pas toujours des résultats satisfaisants dans deux contextes particuliers : la conception de biopuces transcriptomiques lorsque le modèle biologique étudié est complexe (un parasite intracellulaire obligatoire), et d'autre part la conception de biopuces dites phylogénétiques pour l'identification des micro-organismes présents dans un environnement complexe (sol, eau, rumen, intestin…). Le traitement de ces questions peut se découper en trois thèmes :

1. **La problématique de conception des sondes pour biopuces à ADN destinées aux études de transcriptomes.** Dans ce domaine, les logiciels existants sont nombreux, il est même difficile de les dénombrer. Cependant, ils fonctionnent souvent sur le même principe et diffèrent simplement par les critères pris en compte pour la détermination des oligonucléotides. Ces logiciels ont été testés dans le cadre d'une application biologique réelle : l'étude globale de l'expression des gènes d'un parasite eucaryote intracellulaire obligatoire (*Encephalitozoon cuniculi*) au cours de son cycle de développement. Les sondes obtenues présentaient de très nombreux risques d'hybridations croisées et les résultats issus de biopuces « tests » étaient peu satisfaisants. Nous avons donc été amenés à proposer une nouvelle approche pour la conception des sondes.

2. **La problématique de conception des sondes pour biopuces à ADN phylogénétiques.** Par rapport au problème précédent, les similitudes sont nombreuses (les critères à satisfaire pour les sondes sont les mêmes), mais il existe également des particularités : les variations de la séquence d'ARNr 16S (utilisée pour ce type de puces) entre organismes de genres différents sont faibles par rapport aux variations entre les séquences de gènes différents (utilisés pour les puces transcriptomiques). La question de la spécificité des sondes est donc beaucoup plus difficile à résoudre. L'application biologique ayant guidé des recherches est celle de l'identification d'organismes dans un contexte de bioremédiation. Nous proposons un algorithme original de conception de sondes répondant à cette problématique ainsi que son déploiement sur une architecture de type cluster.

3. **Le domaine du Génie Logiciel appliqué aux puces à ADN.** Informaticien de formation, j'ai souhaité, tout en répondant aux questions posées par les biologistes, appliquer les techniques actuelles du Génie Logiciel. Je désirais ainsi apporter une contribution dans un

domaine purement informatique et renforcer le caractère interdisciplinaire de cette thèse. Nous avons fait le constat qu'il existait un manque de composants logiciels réutilisables dans le domaine des puces à ADN en général. Même si des efforts ont été entrepris par la « MGED society » dans le domaine de l'ingénierie des modèles et des ontologies, la problématique de la conception des sondes n'était pas complètement traitée. C'est pourquoi nous avons proposé un « Platform Independant Model » de ce domaine avant d'implémenter un logiciel conforme à ce modèle.

Contributions

Pour mettre en évidence les apports de cette recherche, nous reprenons ici les trois thèmes mentionnés plus hauts.

A l'intérieur du premier thème, les contributions sont d'ordre à la fois biologique et algorithmique. D'une part, nous proposons une approche originale pour la conception d'oligonucléotides pour biopuces ADN. Cette approche, qui consiste à utiliser des sondes composites, permet de gagner en spécificité et en sensibilité de détection des cibles, en particulier lorsque le modèle biologique est complexe. Des premières validations expérimentales ont été réalisées et les résultats sont prometteurs. D'autre part, nous proposons un algorithme permettant de déterminer des sondes conformes à cette approche. Cet algorithme a été utilisé pour concevoir une biopuce spécifique du pathogène *Encephalitozoon cuniculi*.

L'originalité des recherches développées sur le deuxième thème se situe au niveau de l'algorithme proposé. Ce dernier permet de déterminer des sondes spécifiques à un genre donné, dans le cadre d'expériences d'identification d'organismes. Les premières expérimentations biologiques montrent que la spécificité des oligonucléotides est sensiblement améliorée par rapport aux logiciels existants. Nous avons également présenté la mise en place complète du logiciel PhylArray sur une architecture parallèle de type cluster. Ce travail inclut la parallélisation de l'algorithme (parallélisation sur les données), l'intégration de bibliothèques et d'outils de calculs parallèles (MPI, gestionnaire de batch), ainsi que le développement d'une interface Web permettant l'utilisation du logiciel à des utilisateurs préalablement enregistrés.

Dans le domaine du Génie Logiciel, nous avons proposé un modèle du domaine de la conception d'oligonucléotides, en adoptant une démarche conforme à l'approche « Model Driven Architecture » de l'Object Management Group. Cette démarche permettra une maintenance et une réutilisabilité maximum du logiciel proposé. Nous tentons ainsi de démontrer l'intérêt d'une telle approche. En effet, dans les années à venir, le domaine du génie logiciel va être fortement influencé par l'Ingénierie Dirigée par les modèles, qui met les modèles, et non pas les programmes, au centre de la démarche de développement. Il y a de fortes chances que la bioinformatique n'échappe pas à cette évolution, que ce soit en suivant l'architecture MDA ou de l'une de ses variantes.

Perspectives

Dans le cadre de notre nouvelle approche utilisant des sondes composites, des expériences supplémentaires sont bien sûr nécessaires afin de continuer à valider l'approche. La puce destinée à l'étude globale du transcriptome d'*E. cuniculi* à l'aide de ces sondes est actuellement testée. Il serait également intéressant d'utiliser cette technique sur d'autres modèles expérimentaux complexes.

Une des perspectives les plus intéressantes consisterait à coupler la technique des sondes chimériques avec l'algorithme de détermination d'oligonucléotides pour les puces phylogénétiques. Ce dernier permettrait alors de concevoir soit des sondes « classiques », soit des sondes chimériques lorsque des problèmes de spécificité se posent. D'autre part, d'un point de vue logiciel, de nombreuses améliorations peuvent être apportées à l'algorithme de PhylArray. La plus importante serait la possibilité de se placer à n'importe quel niveau dans la taxonomie lorsque l'on souhaite déterminer une sonde. Il serait alors possible de concevoir des oligonucléotides ciblant non plus un genre donné, mais une famille ou un ordre par exemple. Ceci nécessite de coupler à notre base de données de séquences ARNr 16S, une base de données contenant la classification taxonomique sous forme arborescente.

Enfin, une perspective à plus long terme est l'exploitation des résultats produits par les biopuces conçues à l'aide de ces algorithmes. Dans le cadre de processus de bioremédiation, l'utilisation de biopuces capables de cibler des groupes de micro-organismes permettra de connaître les communautés qui jouent potentiellement un rôle dans ces processus. Le fait de pouvoir se placer à des niveaux taxonomiques supérieurs donnera la possibilité d'identifier de nouvelles espèces actives, non encore présentes dans les banques de données d'ARNr.

Au final, ce travail sur la conception de sondes pour puces à ADN ouvre de nombreuses perspectives, et plusieurs voies restent à explorer.

ANNEXE

Introduction à la biologie moléculaire

1 Introduction.

La biologie moléculaire est un domaine de la génétique consacré en grande partie à l'étude des processus fondamentaux qui assurent la pérennité et la transmission de l'information génétique. Cette information représente en quelque sorte les "instructions" nécessaire au développement et au fonctionnement de tout être vivant. Un organisme vivant échange constamment de la matière et de l'énergie avec son environnement, au cours de réactions chimiques extrêmement complexes. Les deux principales molécules entrant en jeu dans ces mécanismes, que l'on pourrait appeler "molécules de la vie", sont les acides nucléiques et les protéines. Schématiquement, les acides nucléiques servent à stocker l'information, et les protéines, fabriquées à partir de cette information, assurent les fonctions nécessaires au fonctionnement des cellules.

2 La cellule, élément de base de l'organisation du vivant.

2.1 Définitions

Une cellule est un compartiment séparé du milieu extérieur par une membrane plasmique. Elle est l'unité de base constituant tout organisme vivant. L'intérieur de la cellule est appelé cytoplasme. On estime à 6.10^{13} le nombre de cellules présentes dans un corps humain, cellules de 320 types différents (peau, muscles, neurones…).

Les organismes vivants sont classés en deux domaines (ou empires) différents : les procaryotes et les eucaryotes.

- les **procaryotes** (absence de noyau vrai) sont des organismes unicellulaires correspondant aux bactéries. Ils doivent présenter une grande adaptabilité au milieu extérieur car ils sont en contact direct avec celui-ci. Les cellules procaryotiques ne comportent pas de compartiments internes complexes tel que des systèmes membranaires développés. Le matériel génétique est situé directement au contact du cytoplasme.
- les **eucaryotes** (présence d'un noyau vrai) regroupent des organismes unicellulaires ou pluricellulaires plus ou moins complexes (Figure 59). Le matériel génétique est situé dans un noyau (présence d'une membrane nucléaire), et le cytoplasme contient de

nombreux compartiments internes jouant un rôle dans le fonctionnement de la cellule (réticulum endoplasmique[21], appareil de Golgi[22], mitochondries[23] …).

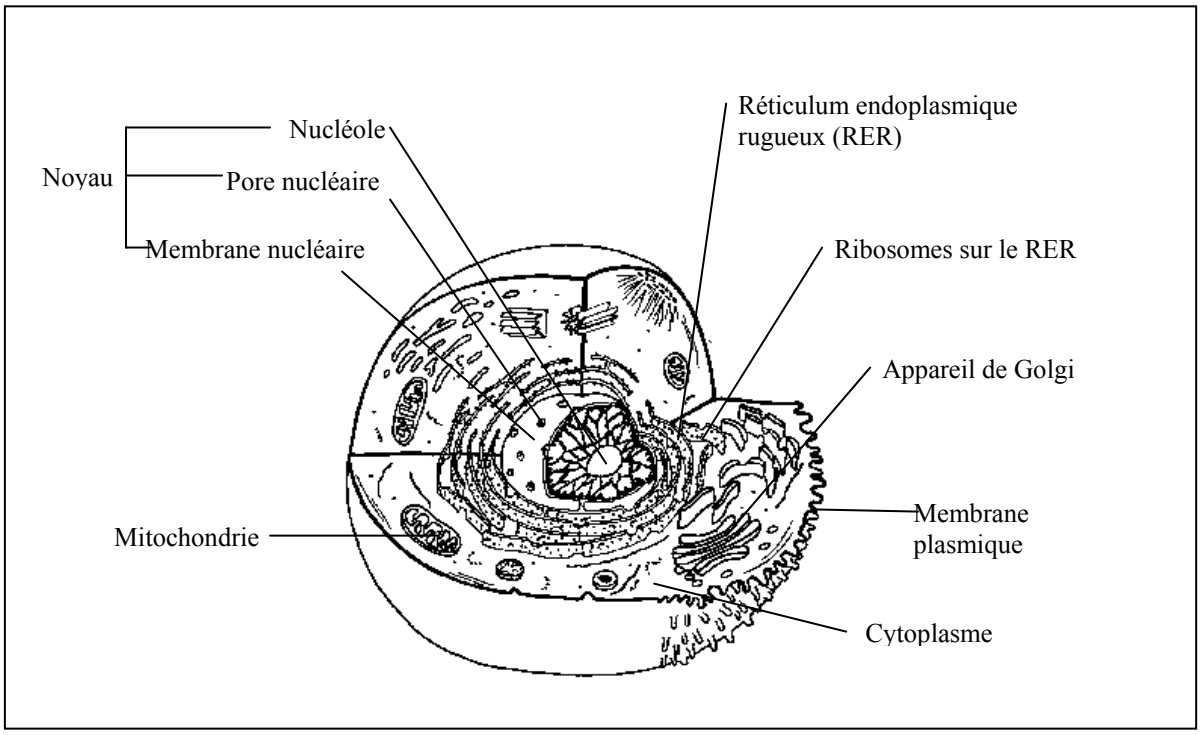


Figure 59 : Schéma d'une cellule eucaryote[24]

2.2 Le métabolisme cellulaire

Pour fonctionner et se développer, la cellule doit sans cesse renouveler ses constituants. De plus, la division cellulaire implique la construction totale de tous les constituants de la cellule. Pour cela, la cellule absorbe des molécules appelées nutriments qu'elle dégrade en molécules plus simples (voie catabolique). C'est à partir de ces molécules plus simples que sont fabriquées les constituants de la cellule (voie anabolique). La Figure 60 illustre ce mécanisme appelé métabolisme cellulaire.

Parmi les molécules entrant en jeu dans le métabolisme cellulaire, on distingue trois grandes catégories : les glucides, les lipides, et les protéines.

Les **glucides** constituent souvent la source d'énergie des cellules : des macromolécules comme le glucose sont dégradées (voie catabolique) par glycolyse pour donner des molécules plus simples comme le pyruvate (Figure 61). Lors de ces réactions, il y a production d'ATP, qui est la molécule apportant l'énergie à toutes les réactions cellulaires.

[21] Système de membranes sur lesquelles se fixent les ribosomes
[22] Organite cellulaire jouant un rôle dans la glycolysation des protéines (entre autres)
[23] Organite cellulaire qui assure la mise en réserve et la production d'énergie de la cellule
[24] Source : J. Soucie © BIODIDAC

Les **lipides** ont principalement deux rôles : ils permettent de constituer des réserves énergétiques pour la cellule d'une part, et d'autre part ils sont les constituants principaux des membranes cellulaires. En effet, cette membrane est formée d'une bicouche lipidique dans laquelle sont enchâssées des protéines qui assurent les échanges avec le milieu extérieur.

Les **protéines** quant à elles sont les molécules de base de tout organisme vivant et font l'objet du chapitre suivant.

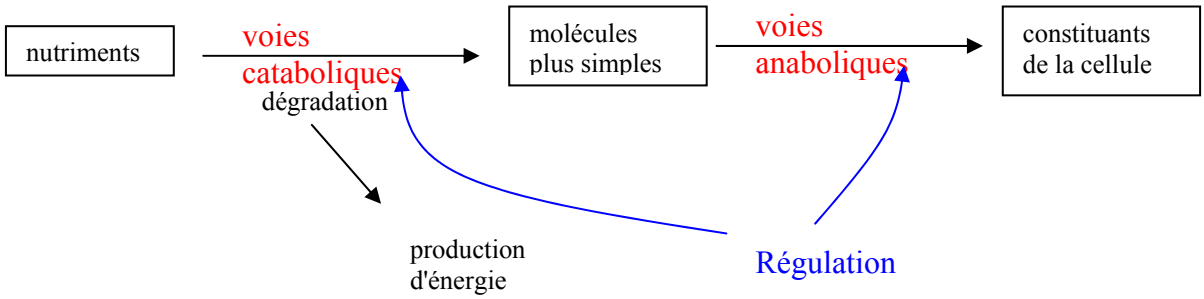


Figure 60 : Le métabolisme cellulaire.

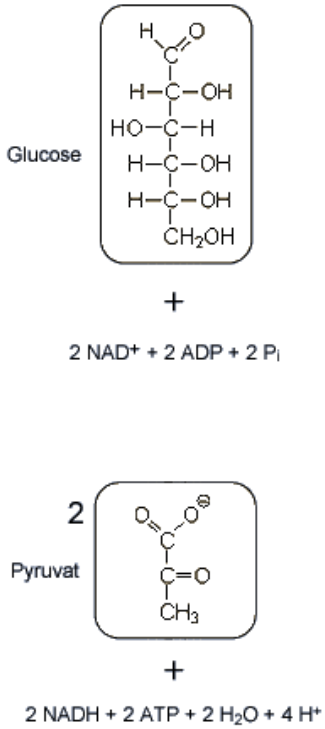


Figure 61 : La glycolyse, dégradation du glucose en pyruvate, avec production d'énergie.

Il existe des mécanismes complexes de régulation des voies métaboliques : des molécules appelées "ligands" agissent sur les enzymes catalysant certaines réactions. Certains ligands peuvent augmenter l'affinité de l'enzyme pour les substrats alors que d'autres, à l'inverse, peuvent diminuer l'affinité de l'enzyme en modifiant sa forme.

2.3 La structure de la membrane cellulaire

Comme nous l'avons vu précédemment, la membrane plasmique est constituée d'une bicouche lipidique dans laquelle sont enchâssées des protéines. Cette structure confère à la cellule une perméabilité sélective : la bicouche lipidique est parfaitement imperméable à la plupart des molécules, mais les protéines peuvent former des pores pour assurer le transport entre les milieux intra et extra cellulaires (voir Figure 62).

Une deuxième fonction des protéines consiste en la réception et la transduction des signaux extérieurs. La partie de la protéine située à l'extérieur de la cellule joue le rôle de récepteur, et lorsqu'une molécule (qui constitue le signal ligand) vient interagir, toute la protéine change de conformation spatiale, y compris la partie située à l'intérieur de la cellule, et donc le signal est transmis au milieu intracellulaire.

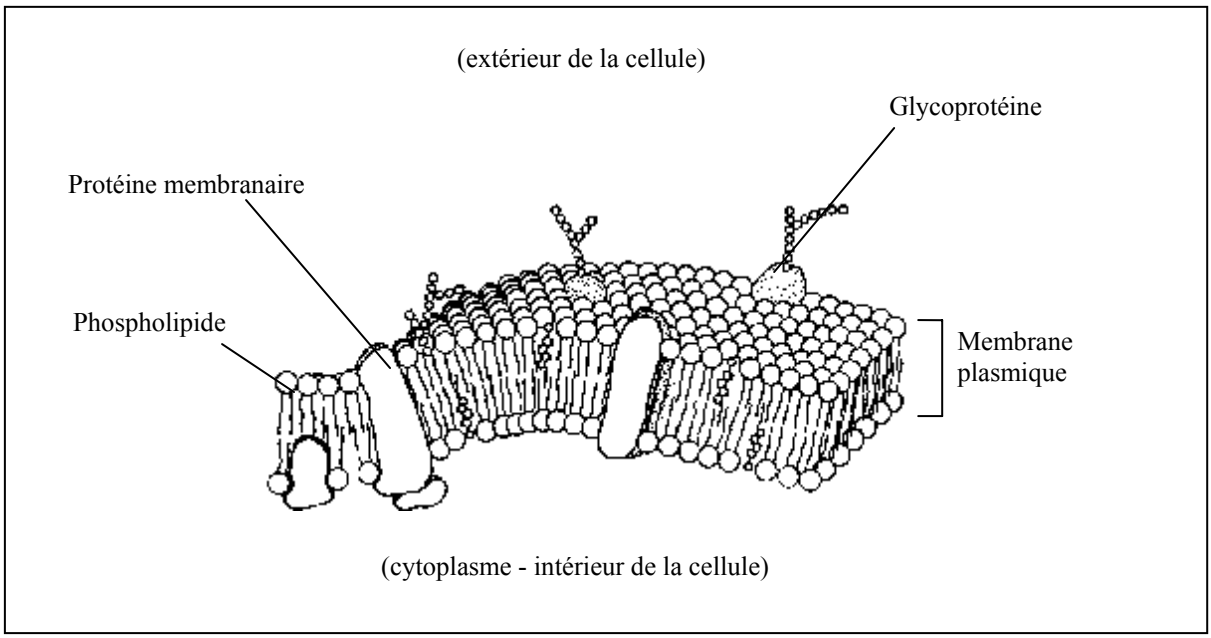


Figure 62 : La membrane plasmique[25].

[25] Source : J. Soucie © BIODIDAC

3 Les protéines.

3.1 Définition

Les protéines sont des macromolécules biologiques complexes essentielles dans la constitution et le fonctionnement de tout être vivant. Une protéine est un polymère d'acides aminés dont l'enchaînement est codé par un gène, et dont la longueur peut aller de 20 à plusieurs centaines d'acides aminés. Cette macromolécule s'organise dans l'espace selon une certaine configuration qui dépend de la séquence des acides aminés (Figure 63).

Le nombre de protéines différentes présentes dans la nature est extrêmement important (sans doute une centaine de milliers), et leurs fonctions au sein des êtres vivants sont multiples :

- la catalyse enzymatique : un catalyseur permet d'accélérer une réaction chimique. Les protéines jouent donc un rôle dans le métabolisme cellulaire.
- le transport et le stockage : l'hémoglobine par exemple, permet le transport de l'oxygène.
- structure : le collagène est une protéine présente dans les tendons.
- la motilité : déplacement, contraction.
- la protection de l'organisme : les anticorps permettent la reconnaissance d'éléments étrangers.
- la régulation : avec des hormones telles que l'insuline.
- la génération et la transmission de signaux.

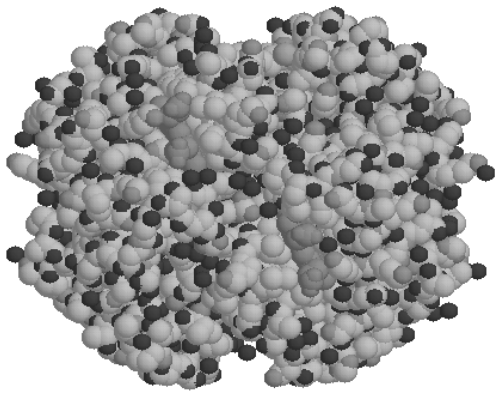


Figure 63 : Molécule d'hémoglobine humaine.

Modèle moléculaire d'une protéine. Représentation des atomes ou rayons de Van der Waals.

3.2 Les acides aminés

Une protéine est constituée d'un enchaînement de molécules simples appelées acides aminés. Un acide aminé est composé d'un atome de carbone central, appelé carbone alpha. A ce carbone alpha sont liés un atome d'hydrogène, un groupe amine (NH2), un groupe carboxyle (COOH), et une chaîne latérale qui diffère selon les acides aminés. Ce groupement peut aller du simple atome d'hydrogène pour la glycine, jusqu'à deux cycles de carbone pour le tryptophane. Dans la nature, on trouve 20 acides aminés différents. La Figure 64 en présente deux exemples et le Tableau 15 présente la nomenclature de codage de l'ensemble des acides aminés.

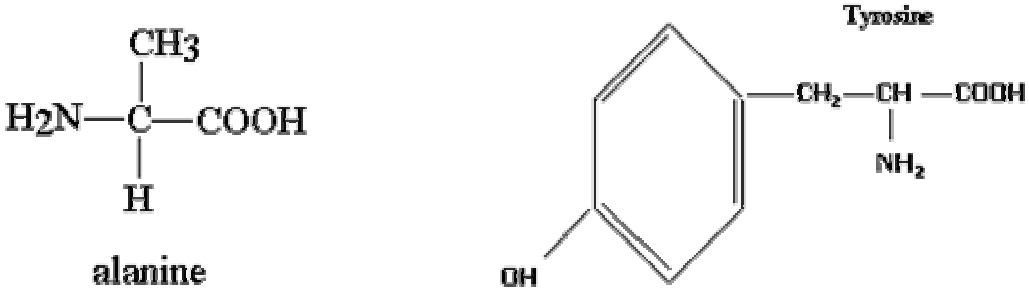


Figure 64 : Exemples d'acides aminés (alanine et tyrosine).

Code à une lettre	Code à 3 lettres	Nom
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic
E	Glu	Glutamic
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophane
Y	Tyr	Tyrosine

Tableau 15 : Code des 20 acides aminés présent dans la nature.

On distingue principalement deux classes d'acides aminés : les acides aminés possédant une chaîne polaire qui ont beaucoup d'interactions avec les molécules d'eau, et les acides aminés possédant une chaîne non polaire qui ont un comportement hydrophobe. Dans une chaîne d'acides aminés, les acides aminés hydrophobes ont tendance à se regrouper à l'intérieur de la macromolécule.

Dans une protéine, les acides aminés sont liés par des liaisons peptides, c'est pour cette raison que les protéines sont également appelées polypeptides. Dans une liaison peptide, l'atome de carbone appartenant au groupe carboxyle de l'acide aminé A_i est lié à l'atome d'azote du groupe amine de l'acide aminé A_{i+1}. Lors de la formation d'une telle liaison, une molécule d'eau est libérée, formée par l'hydrogène, l'oxygène du groupe carboxyle et l'hydrogène du groupe amine (
Figure 65).

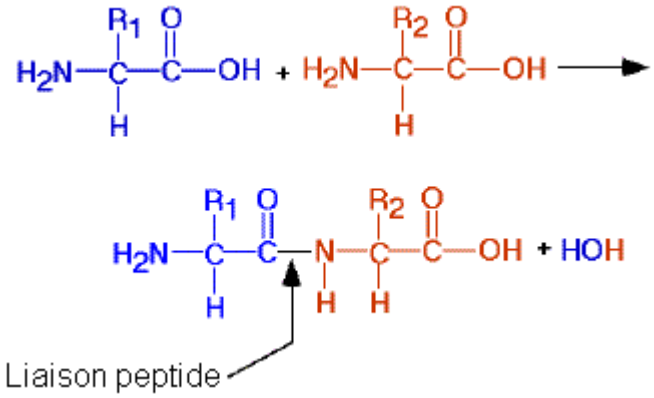


Figure 65 : Formation d'une liaison peptide (R1 et R2 : chaînes latérales des acides aminés).

3.3 Les différents niveaux de structure des protéines

Une protéine ne se résume pas à un simple enchaînement linéaire d'acides aminés. Sa fonction est également déterminée par la structure tridimensionnelle que prend la molécule dans l'espace. C'est pourquoi on distingue plusieurs niveaux dans la structure des protéines.

- La **structure primaire** est l'enchaînement linéaire des acides aminés. Une protéine possède toujours un groupe amine libre à une extrémité et un groupe carboxyle libre à l'autre extrémité. Par convention, on écrit la séquence d'acides aminés en partant du groupe amine libre et allant vers le groupe carboxyle libre.

Exemple :

le polypeptide NH2 – Alanine – Glycine – Tyrosine – COOH
s'écrira avec le code à une lettre : A-G-Y

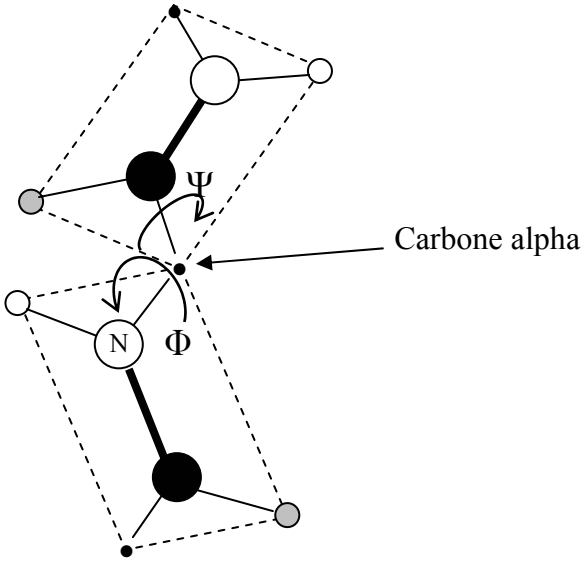


Figure 66 : Illustration de la structure secondaire des protéines.

- La **structure secondaire** est constituée par les structures locales régulières formées par la chaîne d'acides aminés. Dans chacune des liaisons peptidiques, les six atomes mis en jeu (carbone C alpha de l'acide aminé A_i, C, O, N, H, et carbone C alpha de l'acide aminé A_{i+1}) forment un plan. Pour chaque liaison entre ces plans, il existe deux degrés de liberté : les angle Φ et Ψ (voir Figure 66). Les valeurs de ces angles (identiques sur toute une zone de la chaîne) forment la structure secondaire (structure locale). Les structures secondaires que l'on rencontre le plus souvent sont les structures en hélices et les structures en feuillets, qui tendent à minimiser l'état énergétique de la molécule. On trouve également des structures en coude, en tour, et d'autres structures hélicoïdales.
- La **structure tertiaire** est le résultat de repliement de structures secondaires sur elle-même selon par exemple le degré d'hydrophobicité des acides aminés (les acides aminés hydrophobes ont tendance à se regrouper à l'intérieur de la chaîne repliée).
- La **structure quaternaire** est la liaison entre plusieurs sous unités de la protéine qui ont chacune leur structure primaire, secondaire et tertiaire. Elle n'est pas obligatoirement présente chez une protéine.

La Figure 67 présente une représentation graphique de la structure globale d'une protéine.

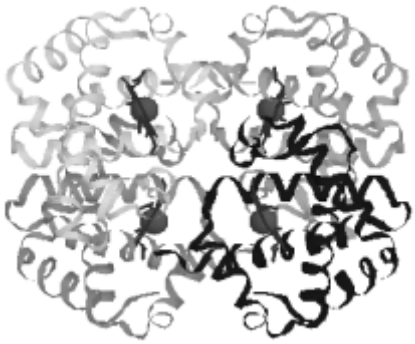


Figure 67 : Illustration de la structure globale de l'hémoglobine humaine.

La structure globale de la protéine (tertiaire) est le résultat du repliement de structures secondaires sur elles-mêmes (représentées par des bandes en hélices).

3.4 Pourquoi est-il intéressant de connaître les séquences et structures des protéines ?

La connaissance de la séquence des acides aminés constituant les protéines est très importante pour :

- l'identification des relations structures-fonctions : la fonction d'une protéine est essentiellement déterminée par sa structure tridimensionnelle.
- connaître l'influence d'une modification des acides aminés dans la séquence. Certaines maladies résultent du changement d'un seul acide aminé dans la séquence d'une protéine;
- la comparaison entre espèce : on compare les séquences des acides aminés d'une même protéine présente chez différentes espèces.
- la production de vaccins : un anticorps est une protéine qui reconnaît une autre protéine (agent infectant), d'où l'utilité de connaître la structure de ces deux protéines

Une question se pose : comment les protéines sont-elle produites par les cellules ? Où intervient la notion de code génétique ? C'est l'objet du chapitre suivant. Les protéines sont produites par une machinerie appelée ribosome. Les acides aminés sont assemblés un par un en suivant une information portée par les acides nucléiques.

4 Les acides nucléiques.

4.1 L'ADN

L'ADN (Acide DésoxyriboNucléique) est une macromolécule enroulée et repliée sur elle-même, qui constitue le support de l'information génétique. Cette information représente en quelque sorte les plans, les instructions nécessaires à la construction et au fonctionnement d'un être vivant. L'ADN est un polymère : c'est une longue chaîne de molécules plus simples appelées nucléotides.

- les nucléotides :

Un nucléotide se compose de trois éléments : un sucre (désoxyribose pour l'ADN), une structure cyclique appelée base, et un groupement phosphate.

Le sucre est composée de cinq carbones numérotés de 1' à 5' (on ajoute le prime pour les distinguer des atomes de carbone de la base). Nous verrons plus loin à quoi sert cette numérotation.

Chaque nucléotide contient l'une des quatre bases suivantes : adénine, guanine, thymine ou cytosine (notées A, G, T et C). Ce sont des molécules complexes qui contiennent des structures cycliques constituées d'atomes de carbone et d'azote. L'adénine et la guanine contiennent deux hétérocycles et sont appelés bases puriques. La cytosine et la thymine contiennent un seul cycle et sont appelées bases pyrimidiques.

- L'ADN, un polynucléotide :

Deux nucléotides peuvent se lier entre eux par une liaison phosphodiester. Le phosphate en 5' d'un nucléotide forme une liaison avec l'hydroxyle en 3' du nucléotide suivant. L'ADN est une immense chaîne de nucléotides (3 milliards chez l'homme) et l'information qu'elle porte est constituée par la suite (ou séquence) des bases de chaque nucléotide, par exemple AGGTTCCCA… Un brin d'ADN possède toujours un groupe 5' phosphate (lié au carbone 5') libre à une extrémité et un groupe 3' hydroxyle libre à l'autre extrémité. Par convention, on écrit toujours une séquence d'ADN dans le sens 5'→3'.

- La double hélice :

La molécule d'ADN existe rarement sous forme de simple chaîne de nucléotides. Le plus souvent, deux chaînes s'enroulent l'une autour de l'autre pour former ce que l'on appelle la double hélice d'ADN (figure 10). L'une des chaîne est dans le sens 5'→3' alors que l'autre est dans le sens 3'→5'. Les bases sont tournées vers l'intérieur de l'hélice et chaque base d'une chaîne est liée à une base de l'autre chaîne (appariement). Une base purique ne peut interagir qu'avec une base pyrimidique, c'est pourquoi on a uniquement des liaisons A-T et G-C. Si l'on connaît la séquence d'un brin de la double hélice, on peut en déduire l'autre, on dit que les deux brins sont complémentaires.

Exemple :

```
brin sens              5'   A T T C G G A T   3'
                            | | | | | | | |
brin complémentaire    3'   T A A G C C T A   5'
```

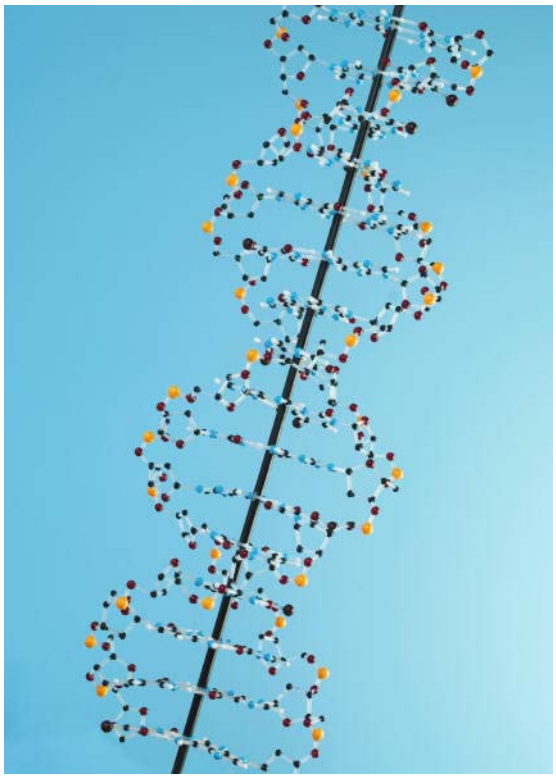


Figure 68 : Représentation schématique de la double hélice d'ADN

4.2 L'ARN

L'ARN (Acide RiboNucléique) est une molécule très semblable à l'ADN, qui entre en jeu notamment pour la synthèse des protéines. Cette molécule d'ARNm (ARN messager), obtenue après la transcription des gènes portés par les chromosomes, sera décodée par les ribosomes pour former les protéines. Il existe d'autres ARN dits fonctionnels (ARNr : ARN ribosomique, ARNt : ARN de transfert…) qui restent sous forme d'ARN et ne sont pas décodés en protéines. Ils assurent donc une fonction en restant sous forme d'ARN. Les ARNr participent à la formation des ribosomes qui sont en réalité des complexes nucléoprotéiques c'est-à-dire formés de protéines et d'acides nucléiques (en l'occurrence d'ARNr). Les ARNt permettent quant à eux d'apporter les acides aminés aux ribosomes pour la synthèse protéique.

Comme l'ADN, l'ARN est un polymère de nucléotides, cependant il existe quelques différences entre les deux molécules. Dans la molécule d'ARN :

- le sucre présent dans chaque nucléotide est un ribose.
- la thymine est remplacée par l'uracile (U).
- la molécule existe le plus souvent sous forme de simple brin et ne forme pas de double hélice.

5 L'expression des gènes.

5.1 Le génome

Dans un organisme vivant, toutes les cellules contiennent une ou plusieurs longues molécules d'ADN organisées en chromosomes. L'ensemble des chromosomes, portant l'information génétique de l'individu, constitue le génome. Toutes les cellules d'un organisme (à de rares exceptions près) contiennent le même génome, ce dernier étant copié à l'identique lors de chaque division cellulaire.

5.2 Qu'est-ce qu'un gène ?

Un gène est une unité d'information et il correspond à un segment d'ADN qui code pour la séquence des acides aminés d'une protéine ou pour des ARN fonctionnels. On estime à 30000 le nombre de gènes chez l'Homme, ils sont dispersés et séparés par des régions inter géniques d'ADN non codant. Les molécules d'ADN ont une capacité énorme de stockage d'information : pour une molécule longue de n bases, le nombre d'ordonnancements possibles est de 4^n. Sachant que chez l'homme, la molécule d'ADN est longue de quelques trois milliards de bases et même si une grande partie de celle-ci est non codante, la quantité d'information stockée reste énorme.

L'information biologique codée dans les gènes est rendue disponible lors de l'expression des gènes. Ce mécanisme, expliqué dans les paragraphes suivants, permet la synthèse de protéines à partir de l'information codée par les gènes. Le fonctionnement d'une cellule dépend de l'activité coordonnée de nombreuses protéines. Ainsi, les mécanismes de régulation d'expression des gènes assurent que les protéines sont synthétisées au bon endroit et au bon moment.

Tous les gènes présents dans une cellule ne sont pas actifs et les différents types cellulaires peuvent exprimer des gènes différents (spécificité tissulaire). L'expression d'un gène est donc régulée par un segment d'ADN en amont de la séquence codante, appelée promoteur. La tendance actuelle est de définir un gène comme étant l'ensemble promoteur + séquence codante.

Chez les procaryotes, toute la séquence suivant le promoteur est codante, et elle est intégralement traduite en protéine. Chez les eucaryotes au contraire, on s'aperçoit que la séquence d'un gène suivant le promoteur peut être découpée en une série de fragments appelées exons (régions codantes) qui sont séparés par des séquences "a priori" non codantes : les introns.

5.3 Le code génétique

On appelle code génétique l'ensemble des règles qui régissent la traduction de la séquence de bases d'un gène en une séquence d'aminés constitutifs d'une protéine. Un acide aminé est codé par un triplet de bases appelé codon. Il existe 64 triplets de bases possibles et seulement 20

acides aminés. La plupart des acides aminés sont donc désignés par plus d'un codon. Cette propriété s'appelle la dégénérescence du code génétique et elle aide à minimiser les effets des mutations. Il existe quelques codons particuliers : TAG, TGA et TAA ne codent pas d'acides aminés mais sont des signaux d'arrêt pour la synthèse protéique, ils sont appelés codons stop. Le codon de la méthionine, ATG, est généralement le signal qui lance la synthèse protéique : on l'appelle codon d'initiation ou codon start. Ainsi, la grande majorité des polypeptides commencent par une méthionine même si celle-ci est quelquefois supprimée ultérieurement (maturation post-traductionnelle). Le Tableau 16 présente le code génétique complet.

	2nd				
1st	**T**	**C**	**A**	**G**	**3rd**
T	F Phe	S Ser	Y Tyr	C Cys	**T**
	F Phe	S Ser	Y Tyr	C Cys	**C**
	L Leu	S Ser	**Ter**	**Ter**	**A**
	L Leu	S Ser	**Ter**	W Trp	**G**
C	L Leu	P Pro	H His	R Arg	**T**
	L Leu	P Pro	H His	R Arg	**C**
	L Leu	P Pro	Q Gln	R Arg	**A**
	L Leu	P Pro	Q Gln	R Arg	**G**
A	I Ile	T Thr	N Asn	S Ser	**T**
	I Ile	T Thr	N Asn	S Ser	**C**
	I Ile	T Thr	K Lys	R Arg	**A**
	M Met	T Thr	K Lys	R Arg	**G**
G	V Val	A Ala	D Asp	G Gly	**T**
	V Val	A Ala	D Asp	G Gly	**C**
	V Val	A Ala	E Glu	G Gly	**A**
	V Val	A Ala	E Glu	G Gly	**G**

Tableau 16 : Le code génétique.

Lorsqu'on dispose d'une séquence d'ADN, il est possible de décrypter la séquence des codons de trois façons différentes. En effet, on n'est pas sûr que la première base commence exactement à la première lettre d'un codon. De plus, du fait de la configuration en double brin complémentaire de l'ADN, on ne sait pas a priori si l'on dispose du brin utilisé pour synthétiser l'ARN (appelé brin matrice) ou de son complémentaire (appelé brin codant car identique à l'ARN synthétisé). Finalement, pour une séquence quelconque, il existe six ensembles de codons possibles et donc six protéines résultantes. On parle de six cadres de lecture.

Exemple :
On dispose de la séquence suivante :
5'…TAATCGAATGGGC…3'
S'il s'agit du brin codant, 3 possibilités permettent le décodage de l'information :
- phase de lecture 1 : TAA TCG AAT GGG
  polypeptide résultant : stop Ser Asn Gly

- phase de lecture 2 : AAT CGA ATG GGC
  polypeptide résultant : Asn Arg Met Gly

- phase de lecture 3 : ATC GAA TGG
  polypeptide résultant : Ile Glu Trp

Le brin dont on dispose peut également être le complémentaire du brin codant, donc encore 3 possibilités (le brin codant est alors : GCCCATTCGATTA, l'ADN étant toujours lu dans le sens 5'→3').

- phase de lecture 1 : GCC CAT TCG ATT
  polypeptide résultant : Ala His Ser Ile

- phase de lecture 2 : CCC ATT CGA TTA
  polypeptide résultant : Pro Ile Arg Leu

- phase de lecture 3 : CCA TTC GAT
  polypeptide résultant : Pro Phe Asp

Lorsqu'on analyse un génome, la première tâche consiste à localiser les codons stop. Toute séquence située entre deux codons stop peut coder potentiellement pour une protéine. Une telle séquence est appelée ORF (Open Reading Frame). Les codons start sont quant à eux plus difficiles à localiser puisqu'ils codent également pour la méthionine et on peut trouver une méthionine n'importe où dans une protéine. Donc finalement les CDS (Coding DNA Sequence, séquence codante d'un gène) sont plus difficiles à identifier que les ORF.

5.4 Le mécanisme de la synthèse protéique

Les protéines sont synthétisées par les cellules en suivant le code porté par la molécule d'ADN. L'ARNm (appelé également transcrit) sert d'intermédiaire entre l'ADN et la protéine. Comme pour la structure des gènes, il existe des différences entre les procaryotes et les eucaryotes au niveau du mécanisme de la synthèse protéique. Chez les eucaryotes, ce mécanisme est beaucoup plus complexe que chez les procaryotes.

- Chez les procaryotes :

La synthèse protéique se déroule en deux phases : transcription et traduction (voir Figure 69).

- la transcription :

C'est la synthèse d'un brin d'ARN à partir d'une matrice d'ADN. L'ARN obtenu est appelé ARN messager (ARNm) car il transmet l'information de la molécule d'ADN jusqu'aux ribosomes qui sont les machineries cellulaires de fabrication des protéines.

Le complexe de transcription (ARN polymérase) est capable de détecter le début d'un gène grâce à des séquences particulières présentes avant chaque zone transcrite appelées séquences promotrices (on rappelle que l'ensemble séquence promotrice + zone transcrite constitue le gène). L'ARN polymérase se déplace le long de la molécule d'ADN et vient se fixer sur la séquence promotrice (Figure 70). Ensuite, il y a « dédoublement » de la double hélice d'ADN,

et synthèse d'un brin d'ARN en utilisant un seul des deux brins d'ADN (appelé brin matrice). La lecture du brin matrice se fait toujours dans le sens 3'→5'. L'ARN est synthétisé à partir du brin matrice en ajoutant au fur et à mesure les bases complémentaires à ce brin, et a donc la même séquence que le brin non matrice (que l'on nomme brin codant).

La fin d'un gène est également matérialisée par une séquence particulière. Lorsque l'ARN polymérase rencontre cette séquence, elle arrête la transcription et se détache de l'ADN.

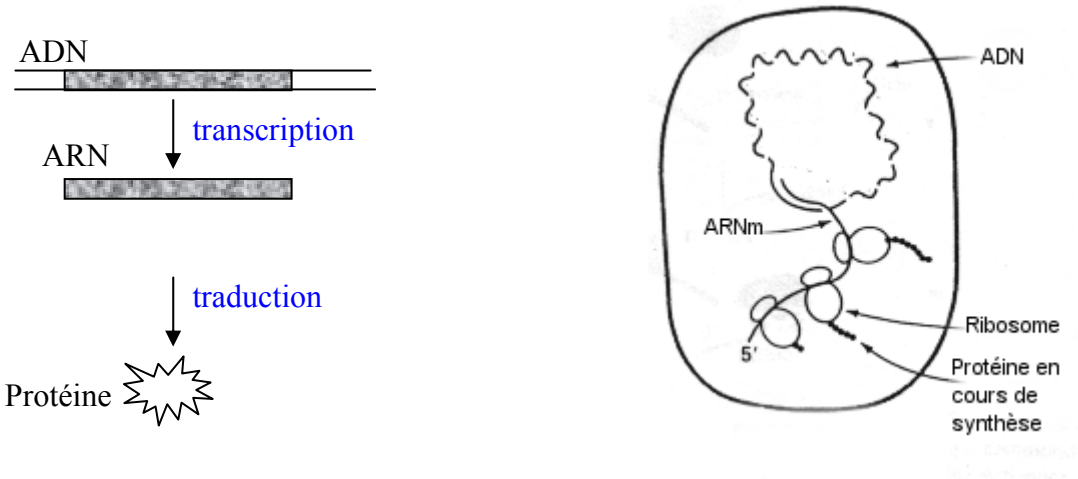


Figure 69 : La synthèse protéique chez les procaryotes : vision schématique (à gauche) et représentation à l'intérieur de la cellule (à droite)

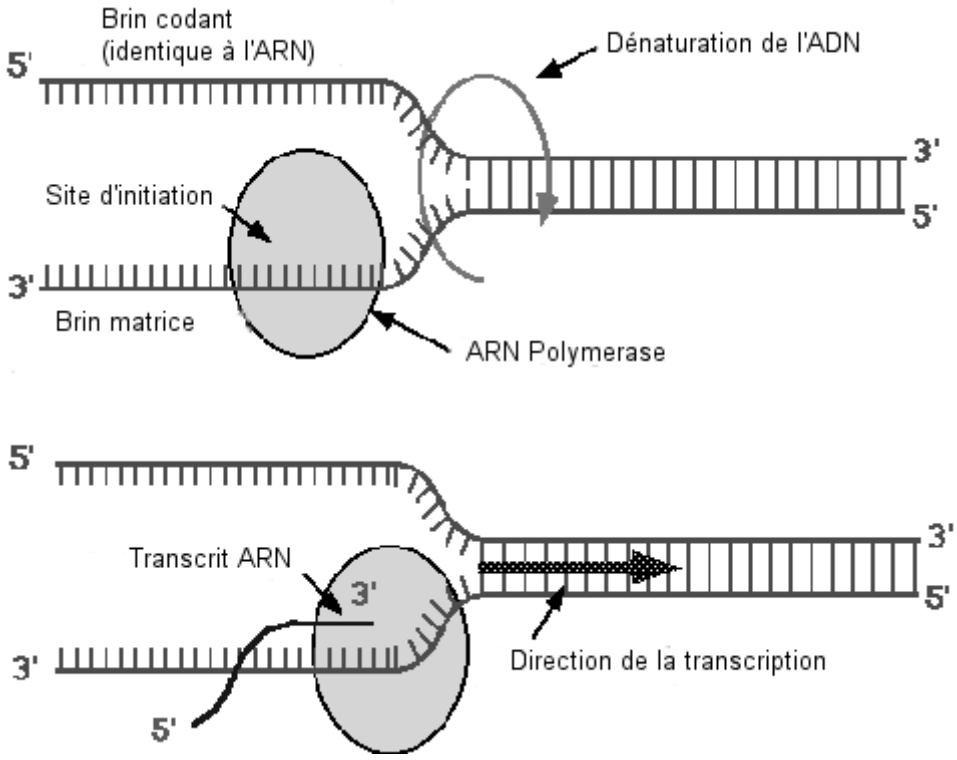


Figure 70 : Principe de la transcription. L'ARN polymérase synthétise un brin complémentaire au brin matrice.

Une cellule n'exprime pas tous ses gènes en même temps, il existe des mécanismes complexes de régulation. Ainsi les bactéries régulent l'expression de leurs gènes de manière à ne produire que ce dont la cellule a besoin. Ceci leur permet de s'adapter aux changements environnementaux. Elles peuvent faire varier la quantité des produits des gènes en contrôlant le niveau de transcription. Parmi les mécanismes de régulation, on peut citer l'induction et la répression.

L'induction permet de stimuler la fabrication d'enzymes impliquées dans des voies métaboliques. Par exemple, la bactérie E. Coli dispose d'un gène codant pour des enzymes permettant de dégrader le lactose (gène lac). Mais elle dispose également d'un gène codant pour une protéine appelée répresseur lac qui vient se fixer sur la séquence promotrice du gène lac, empêchant ainsi la transcription. Dans ces conditions, la bactérie ne produit pas les enzymes nécessaires à la dégradation du lactose. Lorsque la bactérie rencontre du lactose, le petit nombre de molécules des enzymes lac présent dans la cellule permet de commencer à le métaboliser. L'allolactose ainsi produit agit alors comme un inducteur : il vient se fixer à la protéine répresseur lac, changeant sa conformation, et l'empêchant alors de bloquer la transcription du gène lac. La cellule produit alors une grande quantité d'enzymes permettant de métaboliser le lactose.

La répression permet de réguler la synthèse d'un produit (par exemple un acide aminé) : c'est la quantité de produit final qui active ou non la réaction. Lorsque le produit est présent en quantité suffisante dans la cellule, il se fixe à un répresseur, le rendant alors actif. Ce répresseur actif se fixe sur la séquence promotrice du gène codant pour le produit final, empêchant sa transcription.

- la traduction :

Elle met en jeu de petites molécules d'ARN particulières : les ARN de transfert (ARNt). Un ARNt est constitué de deux parties : un triplet de base qui va pouvoir se fixer à l'ARNm par complémentarité, et d'autre part une structure en double hélice sur laquelle est fixé l'acide aminé correspondant. La traduction se déroule dans le cytoplasme et est effectuée par un organite cellulaire, le ribosome, sorte de « machine » à fabriquer les protéines à partir de l'ARN messager.

Le déroulement de la traduction est le suivant : l'ARNm, tel un ruban défile dans le site du ribosome. Pendant ce temps, l'ARNt comportant l'anticodon complémentaire du codon d'ARNm vient se placer, quand le second ARNt arrive et se place, les deux acides aminés forment ensemble une liaison peptidique, démarrant ainsi la chaîne polypeptidique. Puis le ribosome se déplace jusqu'au codon suivant (voir Figure 71).

- Chez les eucaryotes :

On retrouve également les deux phases de transcription et de traduction mais de façon découplée du fait de la présence de la membrane nucléaire. Une étape supplémentaire de maturation de l'ARN messager se produit au niveau nucléaire.

La transcription s'effectue globalement de la même façon chez les eucaryotes et les procaryotes. Toutefois, l'ARN synthétisé par la polymérase, aussi appelé ARN pré-messager, subit une phase de maturation avant son transfert dans le cytoplasme, lieu de la traduction. Cette maturation est accomplie par des protéines spécifiques qui se fixent à l'ARN (addition d'une coiffe en 5', addition d'une queue poly-A en 3' et élimination des introns).

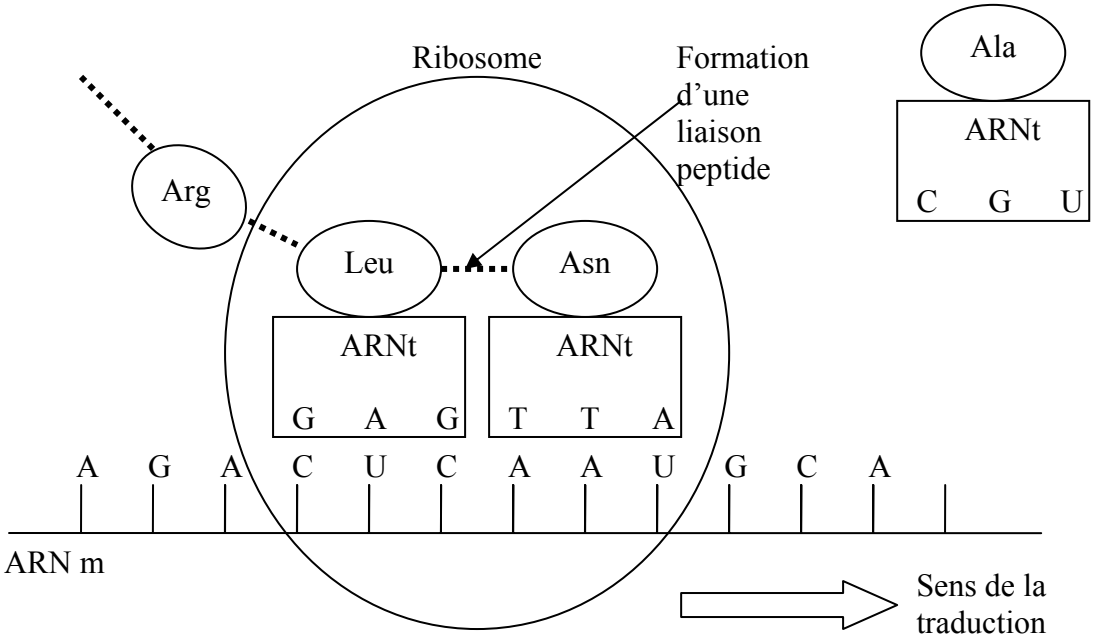


Figure 71 : Principe de la traduction de l'ARN messager en protéine.

Le milieu cellulaire contient tous les types d'ARNt possibles. Le ribosome avance le long de l'ARNm et se place au niveau de deux codons. Les ARNt portant les anticodons correspondants (et donc également les acides aminés correspondants) viennent se fixer sur l'ARNm. Il y a alors formation d'une liaison peptide entre les deux acides aminés. Le ribosome avance ainsi le long de l'ARNm et il y a formation de la chaîne polypeptidique.

- l'**épissage** (Figure 72) :

C'est une phase importante de la maturation de l'ARN pré-messager. Chez les eucaryotes, la séquence d'un gène présente des fragments de séquences non codantes appelées introns. Lors de l'épissage, l'ARN pré-messager est débarrassé de ses introns et la molécule obtenue est l'ARN messager mature.

Il existe également un mécanisme dit d'« épissage alternatif » qui fait que la phase de maturation peut aboutir à plusieurs ARN messagers différents à partir du même pré-messager. En effet, il se peut que tous les introns ne soient pas éliminés. Ce mécanisme est fondamental puisqu'il implique qu'un même gène peut aboutir à la synthèse de plusieurs protéines différentes, permettant ainsi de créer une variabilité génétique.

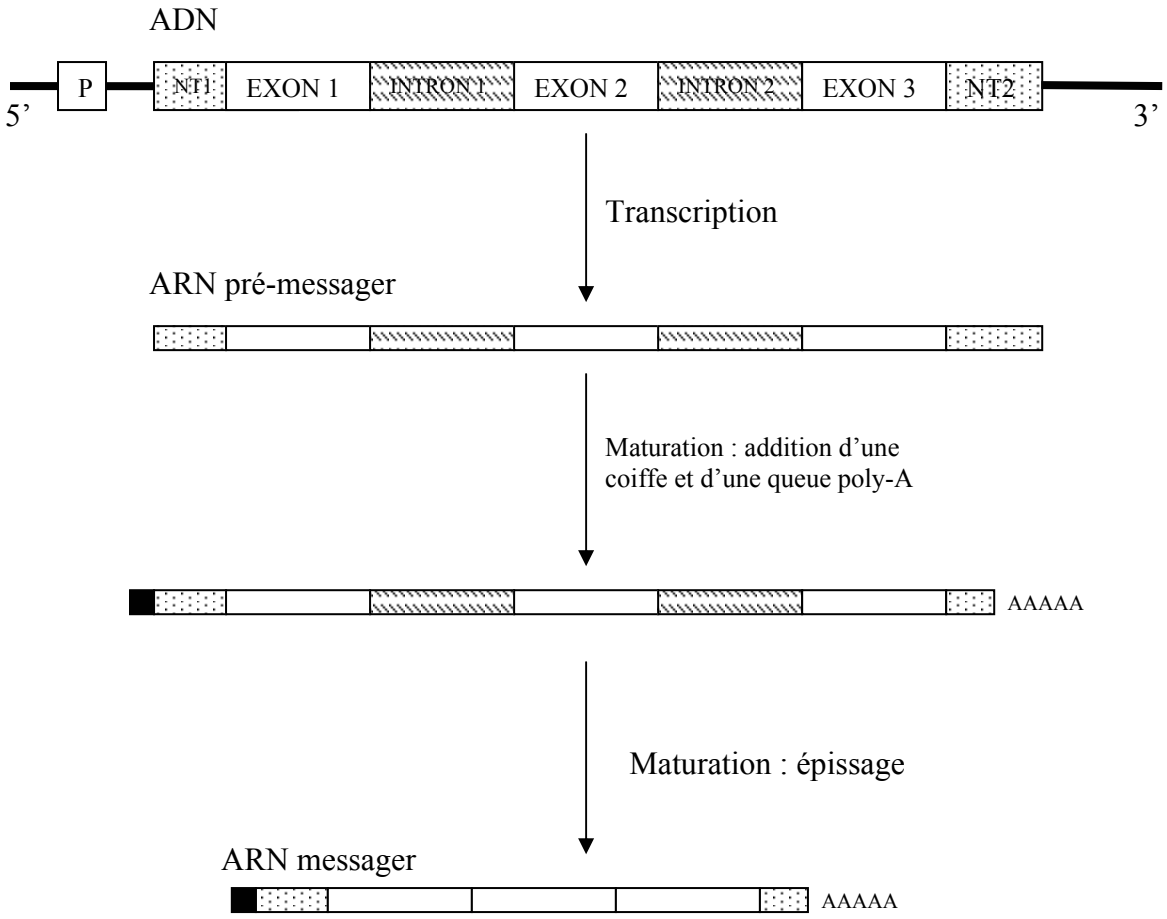


Figure 72 : Maturation de l'ARN messager chez les eucaryotes.

Chez les eucaryotes, un gène est constitué d'une séquence promotrice (P), d'une séquence de tête non traduite (NT1), d'une succession de séquences codantes (exons) et non- codantes (introns) et enfin d'une séquence finale non traduite (NT2). Lors de la transcription, il y a production d'un ARN pré-messager. Lors de la phase la plus importante de la maturation, l'épissage, l'ARN est débarrassé de ses introns.

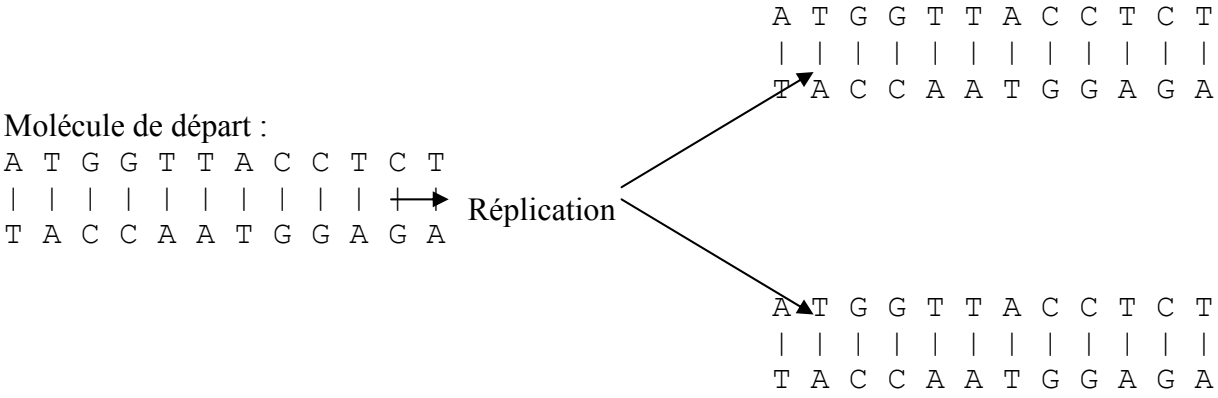
6 Les techniques utilisées pour l'analyse et la compréhension des génomes.

La première étape indispensable à l'analyse du génome d'un organisme est le séquençage : il s'agit de déterminer l'enchaînement des nucléotides de son ADN, sachant que la longueur de cette molécule peut être très importante (environ 3 milliards de nucléotides chez l'homme). Avant de présenter la technique utilisée par les biologistes pour le séquençage, voyons tout d'abord les deux principes à la base de la plupart des techniques d'analyse de génome : la réplication de l'ADN et l'hybridation.

6.1 La réplication de l'ADN

Au sein de la cellule, l'ADN possède une propriété fondamentale, utilisée notamment lors de la division cellulaire : la propriété de se dupliquer. A partir d'une molécule d'ADN double brin, un phénomène appelé réplication permet d'obtenir deux molécules d'ADN (double brin également) identiques à la molécule de départ (c.à.d. possédant le même enchaînement de nucléotides). Le taux d'erreur dans la réplication est de $10^{-9,}$ soit une erreur sur un milliard de nucléotides.

Exemple :



La réplication est assurée par une enzyme : l'ADN polymérase. L'ADN se « dédouble » (il y a séparation des deux brins), cette enzyme se fixe à chacun des brins et ajoute des nucléotides simples présent dans le milieu par complémentarité, puis avance le long de la molécule.

Cependant, deux points importants sont à considérer :

- l'ADN polymérase n'est capable de synthétiser le brin complémentaire que dans le sens 5' 3' (et la lecture du brin matrice se fait donc toujours dans le sens 3' 5').
- l'ADN polymérase ne peut commencer la synthèse qu'en présence d'une amorce : il s'agit d'un petit brin d'ADN long de quelques nucléotides, qui sert à « démarrer » le brin néosynthétisé.

Ainsi, lorsque l'ADN se dédouble progressivement, la synthèse d'un des deux brins se fait de façon continue, alors que la synthèse de l'autre brin se fait par fragments qui sont ensuite recollés.

Exemple :

1. Mise en place des *amorces*

```
       T T A C C T C T 5′
     G   | | |
3′A T G  A T G
  | | |
5′T A C         T C T
     C          | | |
       A A T G G A G A 3′
```

2. Synthèses des **brins complémentaires**

```
       T T A C C T C T 5′
     G   | | | | | | | |
3′A T G  A T G G A G A
  | | |
5′T A C  T A C C T C T
     C   | | | | | | | |
       A A T G G A G A 3′
```

Poursuite de l'écartement des brins d'ADN

3. Pour l'un des deux brins : mise en place d'une nouvelle amorce, pour l'autre : synthèse en continu

```
3′A T G G T T A C C T C T 5′
  | | |     | | | | | | |
  T A C     A T G G A G A

            T A C C T C T
            | | | | | | |
5′T A C C A A T G G A G A 3′
```

4. On obtient deux molécules identiques

```
A T G G T T A C C T C T
| | | | | | | | | | | |
T A C C A A T G G A G A

A T G G T T A C C T C T
| | | | | | | | | | | |
T A C C A A T G G A G A
```

6.2 L'hybridation

L'hybridation est la propriété que possède une molécule d'ADN simple brin de s'associer spontanément avec une autre molécule simple brin lorsque les séquences de base sont complémentaires. Il y a alors formation d'un double brin d'ADN en hélice (voir paragraphe 4.1).

L'hybridation moléculaire est :

- spécifique : sous certaines conditions expérimentales, une séquence d'ADN monobrin ne peut s'apparier qu'à la séquence qui lui est complémentaire dans le génome.
- réversible : l'expérimentateur peut, en jouant sur les conditions expérimentales (essentiellement la température du milieu réactionnel) réaliser ou au contraire supprimer (dénaturation) l'hybridation de deux molécules d'ADN.

Exemple :

Si les deux séquences suivantes sont présentes dans un milieu réactionnel dans certaines conditions de température :

```
A T C G T C A G C T A              T A G C A G T C G A T
```

il y aura formation d'une molécule double brin :

```
A T C G T C A G C T A
| | | | | | | | | | |
T A G C A G T C G A T
```

6.3 Le séquençage

Le séquençage est l'opération qui consiste à déterminer la séquence des bases d'un fragment d'ADN donné. La technique a été mise au point par Sanger en 1977. Elle repose sur deux principes essentiels :

- la possibilité de séparer des fragments d'ADN de longueurs différentes par électrophorèse : si l'on dispose plusieurs fragments d'ADN au même endroit sur un gel et qu'on les place dans un champ électrique pour les faire migrer, plus un fragment est petit, plus il se déplacera rapidement et donc plus il ira loin sur le gel. Ce système est suffisamment résolutif pour séparer deux brins dont la longueur diffère d'une seule base.
- le phénomène de réplication de l'ADN peut fonctionner in vitro : si l'on met en présence de l'ADN polymérase, un brin d'ADN muni d'une amorce adéquate et des nucléotides « libres », l'enzyme synthétise le brin complémentaire.

Sanger a également mis au point des nucléotides dit « terminateur » : lorsque l'un d'entre eux est ajouté au brin néosynthétisé, l'ADN polymérase arête la synthèse, elle ne peut plus ajouter de nucléotide. Il s'agit en réalité de didésoxynucléotides ne présentant pas de groupement hydroxyle en 3'. L'absence de ce groupement ne permet plus la formation d'une liaison phosphodiester et la synthèse d'ADN est alors stoppée.

Le principe du séquençage est le suivant : une ADN polymérase va effectuer des copies du brin d'ADN à séquencer, mais ces copies seront incomplètes, la synthèse va s'arrêter aléatoirement au niveau d'une base donnée avec insertion d'un terminateur.

Le fragment d'ADN à séquencer est placé dans un milieu réactionnel contenant :

- l'amorce à partir de laquelle la synthèse de la copie sera réalisée
- les 4 nucléotides (dA, dT, dC, dG),
- les 4 terminateurs (ddA, ddT, ddC, ddG) marqués chacun par un fluorochrome distinct
- une ADN polymérase

L'ensemble est soumis à une succession de cycles de polymérisation au cours desquels l'ADN polymérase peut, au niveau de chaque nucléotide de l'ADN matrice, incorporer un nucléotide classique ou un terminateur. Dans le cas où elle incorpore un nucléotide classique, la synthèse peut continuer, dans le cas contraire elle s'arrête. Ce choix étant aléatoire, chaque base de l'ADN matrice aura statistiquement vu un certain nombre de fois l'incorporation d'un terminateur, si bien que le milieu réactionnel contient l'ensemble des molécules néosynthétisées possibles. Ces molécules sont ensuite dénaturées, puis migrées dans un gel d'électrophorèse afin d'être séparées selon leur taille. On peut ainsi reconstituer la séquence en analysant la nature du fluorochrome terminant chacun de ces fragments néosynthétisés, du plus petit (premier nucléotide de la matrice) au plus grand (dernier nucléotide de la matrice).

6.4 La PCR (Polymerase Chain Reaction)

La réaction de polymérisation en chaîne est une technique qui permet d'obtenir de multiples copies d'une séquence d'ADN. Le matériel de départ est le suivant :

- une molécule d'ADN double brin
- deux amorces délimitant la séquence à amplifier
- une ADN polymérase particulière : la Taq polymérase, qui est capable de synthétiser de l'ADN à haute température (65-75°C)

La réaction est constituée d'une succession de cycles (voir Figure 73), chaque cycle étant constitué des étapes suivantes :

- Dénaturation de l'ADN : chauffage du mélange pour séparer les brins d'ADN (~95°C)
- Hybridation des amorces : refroidissement pour permettre aux amorces de s'hybrider sur les brins d'ADN
- Extension des amorces : chauffage (65°C-75°C) pour que la Taq polymérase synthétise les brins complémentaires d'ADN à partir des amorces

A chaque cycle, le nombre de brin d'ADN double, donc le rendement de la réaction est exponentiel, ce qui permet d'obtenir très rapidement des millions de copies de la séquence de départ.

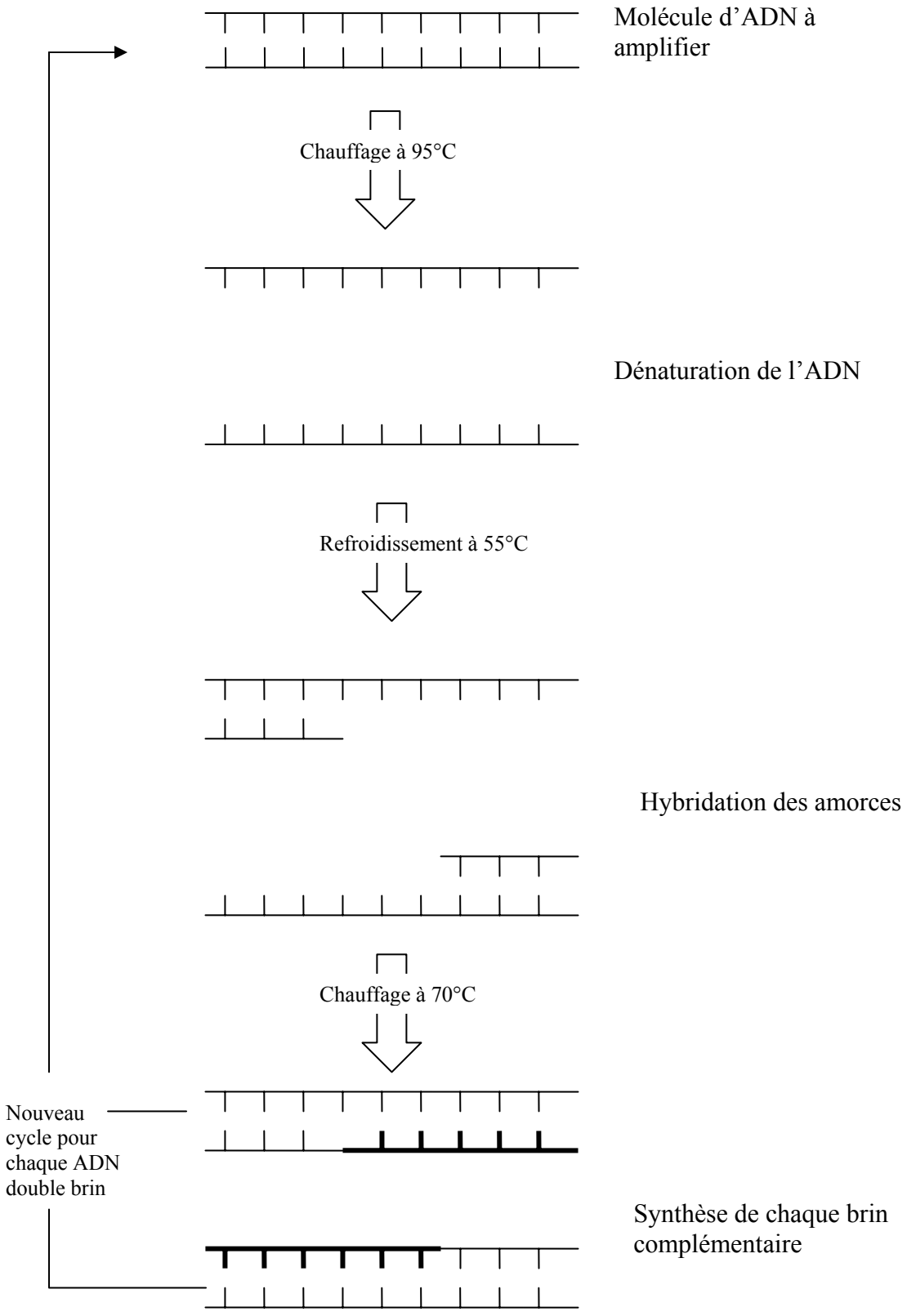


Figure 73 : Les différentes étapes d'un cycle d'une expérience PCR.

Bibliographie

Albertson, D.G., Collins, C., McCormick, F. and Gray, J.W. (2003) Chromosome aberrations in solid tumors, Nat. Genet., 34:369-376.

Altschul, S., Gish, W. M. W., Myers, E. W. and Lipman, D. (1990) A basic local alignment search tool, J. Mol. Biol., 215:403-410.

Altschul, S.F., Madden,T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res., 25:3389-3402.

Amann, R.I., Ludwig, W. and Schleifer, K.H. (1995) Phylogenetic identification and in situ detection of individual microbial cells without cultivation, Microbiol Rev., 59:143-169.

Anderson, J.A. and Rosenfeld, E. (eds) (1998) Neurocomputing: Foundations of Research, MIT Press.

Ashelford, K.E., Weightman, A.J. and Fry, J.C. (2002) PRIMROSE: a computer program for generating and estimating the phylogenetic range of 16S rRNA oligonucleotide probes and primers in conjunction with the RDP-II database, Nucleic Acids Res., 30:3481-3489.

Atkinson, C. (2004) Unifying MDA and Knowledge Representation Technologies, University of Mannheim, edoc 2004.

Azuaje, F. (2001) A computational neural approach to support the discovery of gene function and classes of cancer, IEEE Trans Biomed Eng., 48:332-339.

Baldi, P. and Long, A.D. (2001) A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes, Bioinformatics, 17:509-519.

Ball, C.A., Brazma, A., Causton, H., Chervitz, S., Edgar, R., Hingamp, P., Matese, J.C., Parkinson, H., Quackenbush, J., Ringwald, M. et al (2004) Submission of microarray data to public repositories, PLoS Biol., 2:e317, 1276-1277.

Barra, V. (2004) Analysis of gene expression data using functional principal components, Comput Methods Programs Biomed., 75:1-9..

Barra, V. (2006) Robust segmentation and analysis of DNA microarray spots using an adaptative split and merge algorithm, Comput Methods Programs Biomed., 81:174-180.

Barrett, M.T. (2005) A decade of genome-wide biology, Nature Genetics, 37, S3.

Behr, M.A., Wilson, M.A., Gill, W.P., Salamon, H., Schoolnik, G.K., Rane, S. and Small, P.M. (1999) Comparative genomics of BCG vaccines by whole-genome DNA microarray, Science, 284:1520-1523.

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. and Wheeler, D.L. (2006) GenBank, Nucleic Acids Res., 34(Database issue):D16-20.

Benton, D., Konnerth, K. and Markel, S. (2000) The OMG Life Sciences Research Domain Task Force, ACM SIGBIO Newsletter, 20:14-21.

Bézivin, J. (2004) In search of a Basic Principle for Model Driven Engineering, Upgrade, V:21-24.

Bézivin, J., Hammoudi, S., Lopes, D. and Jouault, F. (2004) Applying MDA Approach for Web Service Platform, Proceedings of 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004), pages 58-70.

Bilban, M., Buehler, L.K., Head, S., Desoye, G. and Quaranta, V. (2002) Normalizing DNA microarray data, Curr Issues Mol Biology 2002, 4:57-64.

Binder, H., Kirsten, T., Loeffler, M., and Stadler, P.F. (2004) Sensitivity of Microarray Oligonucleotide Probes: Variability and Effect of Base Composition, J. Phys. Chem. B, 108:18003 -18014.

Borneman, J., Chrobak, M., Della Vedova, G., Figueroa, A. and Jiang, T. (2001) Probe selection algorithms with applications in the analysis of microbial communities, Bioinformatics, 17 Suppl 1:S39-48.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C. et al (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data, Nat Genet., 29:365-71.

Brown, M.P., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M. and Haussler, D. (1997) Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. USA, 97: 262-267.

Buck, M.J. and Lieb, J.D. (2004) ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments, Genomics, 83:349-60.

Burges, C.J.C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 2:121-167.

Celis, J.E., Kruhoffer, M., Gromova, I., Frederiksen, C., Ostergaard, M., Thykjaer, T., Gromov, P., Yu, J., Palsdottir, H., Magnusson, N., Orntoft, T.F. (2000) Gene expression profiling: monitoring transcription and translation products using DNA microarrays and proteomics, FEBS Lett., 480:2-16.

Chen, H. and Sharp, B.M.. (2002) Oliz, a suite of Perl scripts that assist in the design of microarrays using 50mer oligonucleotides from the 3' untranslated region., BMC Bioinformatics, 3:27.

Chin, K.V. and Kong, A.N. (2002) Application of DNA microarrays in pharmacogenomics and toxicogenomics, Pharm Res., 19:1773-1778.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O. and Herskowitz, I. (1998) The transcriptional program of sporulation in budding yeast, Science, 282:699-705.

Cochrane, G., Aldebert, P., Althorpe, N., Andersson, M., Baker, W., Baldwin, A., Bates, K., Bhattacharyya, S., Browne, P., van den Broek, A. et al (2006) EMBL Nucleotide Sequence Database: developments in 2005, Nucl. Acids Res., 34(Database issue):D10-D15.

Cole, J.R., Chai, B., Farris, R.J., Wang, Q., Kulam, S.A., McGarrell, D.M., Garrity, G.M. and Tiedje, J.M. (2005) The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis, Nucleic Acids Res., 33(Database issue):D294-6.

Crowther, D.J. (2002) Applications of microarrays in the pharmaceutical industry, Current Opinion in Pharmacology, 2:551-554.

Curry, A. and Smith, H.V. (1998) Emerging pathogens: Isospora, Cyclospora and microsporidia, Parasitology, 117 Suppl: S143-159.

De Smet, F., Mathys, J., Marchal, K., Thijs, G., De Moor, B., Moreau, Y. (2002) Adaptive quality-based clustering of gene expression profiles, Bioinformatics, 18:735-746.

Debouck, C. and Goodfellow, P.N. (1999) DNA microarrays in drug discovery and development, Nat. Genet., 21:48-50.

DeRisi, J.L., Iyer V.R. and Brown, P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale, Science, 278:680-686.

DeSantis, T.Z., Dubosarskiy, I., Murray, S.R. and Andersen, G.L. (2003) Comprehensive aligned sequence construction for automated design of effective probes (CASCADE-P) using 16S rDNA, Bioinformatics, 19:1461-1468.

Ebedes, J. and Datta, A. (2004) Multiple sequence alignment in parallel on a workstation cluster, Bioinformatics, 20:1193-1195.

Eisen, M., Spellman, P., Brown, P. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, Proc Natl Acad Sci USA, 95:14863-14868.

EMBL (2000) Gene Expression RFP Response. Initial Submission from EMBL-EBI (European Bioinformatics Institute). Version 1.0. OMG document # lifesci/2000-11-16.

Favre, J.M. (2004) Towards a Basic Theory to Model Driven Engineering, Proceedings of UML 2004 – Workshop in Software Model Engineering (WISME 2004).

Felske, A., Rheims, H., Wolterink, A., Stackebrandt, E. and Akkermans, A.D. (1997) Ribosome analysis reveals prominent activity of an uncultured member of the class Actinobacteria in grassland soils, Microbiology, 143:2983-2989.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M. and Klein, M. (2000) OIL in a nutshell In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, Springer-Verlag.

Fox, G.E., Stackebrandt, E., Hespell, R.B., Gibson, J., Maniloff, J., Dyer, T.A., Wolfe, R.S., Balch, W.E., Tanner, R.S., Magrum, L.J. et al. (1980) The phylogeny of prokaryotes, Science, 209:457-463.

Franco, C. et Maddeh, A. (2006) Développement d'une application extranet pour le logiciel PhylArray fonctionnant sur le cluster de l'ISIMA, Rapport de projet 2ème année, ISIMA(Institut Superieur d'Informatique de Modelisation et de leurs Applications).

Fuhrman, S., Cunningham, M.J., Wen, X., Zweiger, G., Seilhamer, J.J. and Somogyi, R. (2000) The application of shannon entropy in the identification of putative drug targets, Bio Systems, 55:5-14.

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M. and Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics, 16:906-914.

Gamma, E., Helm, R., Johnson, R. and Vlissides J. (1994) Design Patterns Elements of reusable object- oriented sofware, Addison-Wesley.

Gene Ontology Consortium (2004) The Gene Ontology (GO) database and informatics resource, Nucleic Acids Res, 32: Database issue D258-D261.

Gokhale, A., Natarjan, B., Schmidt, D.C., Nechypurenko, A., Wang, N., Gray, J., Neema, S., Bapty, T. and Parsons, J. (2002) CoSMIC: An MDA Generative Tool for Distributed Real-time and Embdedded Component Middleware and Applications, Proceedings of the OOPSLA 2002 Workshop on Generative Techniques in the Context of Model Driven Architecture.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, 286:531-537.

Gordon, P.M. and Sensen, C.W. (2004) Osprey: a comprehensive tool employing novel methods for the design of oligonucleotides for DNA sequencing and microarrays, Nucleic Acids Res., 32:e133.

Gray, J., Zhang, J., Lin, Y., Wu, H., Roychoudhury, S., Sudarsan, R., Gokhale, A., Neema, S., Shi, F. and Bapty, T. (2004) Model-Driven Program Transformation of a Large Avionics Framework, Generative Programming and Component Engineering (GPCE 2004), pages 361-378.

Gruber, T.R. (1993) A translation approach to portable ontology specifications, Knowledge Acquisition Journal, 5:199-220.

Gruber, T.R. (1995) Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human-Computer Studies, 43:907-928.

Guarino, N. and Giaretta, P. (1995) Ontologies and knowledge bases: Towards a terminological clarification. In Mars, N., editor, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing 1995, pages 25-32, Amsterdam, NL.

Gusev, V.D., Nemytikova, L.A. and Chuzhanova, N.A. (1999) On the complexity measures of genetic sequences, Bioinformatics, 15:994-999.

Gusfield, D. (1997) Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press.

Gushin, D.Y., Mobarry, B.K., Proudnikov, D., Stahl, D.A, Rittmann, B.E. and Mirzabelov, A. (1997) Oligonucleotide microchips as genosensors for determinative and environmental studies in microbiology, Appl. Environ. Microbiol., 63:2397-2402.

Guthke, R., Schmidt-Heck, W., Hahn, D. and Pfaff, M. (2001) Gene Expression Data Mining for Functional Genomics using fuzzy technology, In International Series in Intelligent Technologies: Advances in Computational Intelligence and Learning: Methods and Applications, Kluwer Academic Publishers.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) Gene Selection for Cancer Classification using Support Vector Machines, Machine Learning, 46: 389-422.

Hancock, J.M. (2002) Genome size and the accumulation of simple sequence repeats: implications of new data from genome sequencing projects, Genetica, 115:93-103.

Haykin, S. (1994) Neural Networks, a Comprehensive Foundation, Macmillan, New York, NY.

Hazelhurst, S., Liptak, Z. and Zimmerman, J. (2003) A Comparative Study of Biological Distances for EST Clustering, Technical Report TR-Wits-CS-2003.

Hegde, P., Qi, R., Gaspard, R., Abernathy, K., Dharap, S., Earle-Hughes, J., Gay, C., Nwokekeh, N.U., Chen, T., Saeed, A.I., Sharov, V., Lee, N.H., Yeatman, T.J., Quackenbush, J. (2001) Identification of tumor

markers in models of human colorectal cancer using a 19,200-element complementary DNA microarray, Cancer Res., 61:7792-7797.

Herrero, J., Valencia, A. and Dopazo, J. (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns, Bioinformatics, 17:126-136.

Hertz, J., Krogh, A. and Palmer, R.G. (1991) An introduction to the theory of Neural Computation, Addison-Wesley.

Heyer, L.J., Kruglyak, S. and Yooseph, S. (1999) Exploring expression data: identification and analysis of coexpressed genes, Genome Res., 9: 1106-1115.

Hill, D., Rimour, S., Peyret, P. (2002) MAGE compliant Object Model for Oligonucleotide Design in Microarray experiments: Application to the eukaryotic parasite Encephalitozoon cuniculi, Proceedings of Objects in Bio- & Chem-Informatics, OMG Object Management Group, Washington, DC (USA), pp. 39-44.

Hirschhorn, J.N., Sklar, P., Lindblad-Toh, K., Lim, Y.M., Ruiz-Gutierrez, M., Bolk, S., Langhorst, B., Schaffner, S., Winchester, E. and Lander, E.S. (2000) SBE-TAGS: an arraybased method for efficient single-nucleotide polymorphism genotyping, Proc. Natl Acad. Sci. USA, 97:12164-12169.

Horrocks, I., Patel-Schneider, P.F. and van Harmelen, F. (2002) Reviewing the Design of {DAML+OIL}: An Ontology Language for the Semantic Web, Proceedings of the 18th Nat. Conf. on Artificial Intelligence (AAAI~2002).

Hugenholtz, P. and Pace, N.R. (1996) Identifying microbial diversity in the natural environment : a molecular phylogenetic approach, Tibtech., 14:190-197.

Ishkanian, A.S., Malloff, C.A., Watson, S.K., DeLeeuw, R.J., Chi, B., Coe, B.P., Snijders, A., Albertson, D.G., Pinkel, D. and Marra, M.A. (2004) A tiling resolution DNA microarray with complete coverage of the human genome, Nat. Genet., 36:299-303.

Ivanov, I., Schaab, C., Planitzer, S., Teichmann, U., Machl, A., Theml, S., Meier-Ewert, S., Seizinger, B., Loferer, H. (2000) DNA microarray technology and antimicrobial drug discovery, Pharmacogenomics, 1:169-78.

Iyer, V.R., Horak, C.E., Scafe, C.S., Botstein, D., Snyder, M. and Brown, P.O. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF, Nature, 409:533-538.

Jain, A.K. and Dubes, R.C. (1998) Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, USA.

Kallioniemi, A., Kallioniemi, O.P., Sudar, D., Rutovitz, D., Gray, J.W., Waldman, F. and Pinkel, D. (1992) Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors, Science, 258:818-821.

Kane, M.D., Jatkoe, T.A., Stumpf, C.R., Lu, J., Thomas, J.D. and Madore, S.J. (2000) Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays, Nucleic Acids Res., 28:4552-4557.

Katinka, M.D., Duprat, S., Cornillot, E., Metenier, G., Thomarat, F., Prensier, G., Barbe, V., Peyretaillade, E., Brottier, P., Wincker, P., Delbac, F., El Alaoui, H., Peyret, P., Saurin, W., Gouy, M., Weissenbach, J. and Vivares, C.P. (2001) Genome sequence and gene compaction of the eukaryote parasite Encephalitozoon cuniculi, Nature, 414: 450-453.

Kennedy, G.C., Matsuzaki, H., Dong, S., Liu, W.M., Huang, J., Liu, G., Su, X., Cao, M., Chen, W., Zhang, J., Liu, W., Yang, G., Di, X., Ryder, T., He, Z., Surti, U., Phillips, M.S., Boyce-Jacino, M.T., Fodor, S.P. and Jones, K.W. (2003) Largescale genotyping of complex DNA, Nat. Biotechnol., 21:1233-1237.

Kerr, M.K. and Churchill, G.A. (2001) Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments, Proc Natl Acad Sci USA, 98:8961-8965.

Kerr, M.K., Martin, M. and Churchill, G.A. (2000) Analysis of variance for gene expression microarray data, Journal of Computational Biology, 7:819-837.

Kohonen, T. (1997) Self-Organizing Maps, Springer, Berlin, 1997.

Kolmogorov, A.N. (1965) Three approaches to the quantitative definition of information, Probl. Inform. Transmis., 1:1-7.

Kudoh, K., Ramanna, M., Ravatn, R., Elkahloun, A.G., Bittner, M.L., Meltzer, P.S., Trent, J.M., Dalton, W.S. and Chin, K.V. (2000) Monitoring the expression profiles of doxorubicin-induced and doxorubicin-resistant cancer cells by cDNA microarray, Cancer Res., 60:4161-4166.

Kuhn, K.M., DeRisi, J.L., Brown, P.O. and Sarnow, P. (2001) Global and specific translational regulation in the genomic response of Saccharomyces cerevisiae to a rapid transfer from a fermentable to a nonfermentable carbon source, Mol Cell Biol., 21:916-927.

Lars, J. (2002) Cadmium overload and toxicity, Nephrol. Dial. Transpl., 17:35-39.

Lemkin, P.F., Thornwall, G.C., Walton, K.D. and Hennighausen, L. (2000) The microarray explorer tool for data mining of cDNA microarrays: application for the mammary gland, Nucleic Acids Res., 28:4452-4459.

Lempel, A. and Ziv, J. (1976) On the complexity of finite sequences. IEEE Trans. Inf. Theory, IT-22:75-81.

Li, F. and Stormo, G.D. (2001) Selection of optimal DNA oligos for gene expression arrays, Bioinformatics, 17:1067-1076.

Li, K.-B. (2003) ClustalW-MPI: ClustalW analysis using distributed and parallel computing, Bioinformatics, 19:1585-1586.

Lieb, J.D., Liu, X., Botstein, D. and Brown, P.O. (2001) Promoterspecific binding of Rap1 revealed by genome-wide maps of protein-DNA association, Nat. Genet., 28:327-334.

Lipman, D. and Pearson, W.R. (1985) Rapid and sensitive protein similarity searches, Science, 227:1435-1441.

Ludwig, W., Strunk, O., Westram, R., Richter, L., Meier, H., Yadhukumar, Buchner, A., Lai, T., Steppi, S., Jobb, G. et al. (2004) ARB: a software environment for sequence data, Nucleic Acids Res, 32:1363-1371.

Manber, U. and Myers, G. (1993) Suffix arrays: A new method for on-line string string searches, SIAM Journal on Computing, 22:935-948.

Meinkoth, J. and Wahl, G.M. (1984) Hybridization of Nucleic Acids Immobilized on Solid Supports, Analytical Biochemistry, 138: 267-284.

Mikhailov, D., Cofer, H. and Gomperts, R. (2001) Performance optimization of ClustalW: parallel ClustalW, HT Clustal, and MULTICLUSTAL. White papers, Silicon Graphics, Mountain View, CA.

Navarro, G. (2001) A guided tour to approximate string matching, ACM Computing Surveys, 33:31-88.

Nielsen, H.B., Wernersson, R. and Knudsen, S. (2003) Design of oligonucleotides for microarrays and perspectives for design of multi-transcriptome arrays, Nucleic Acids Res., 31:3491-3496.

Object Management Group (2001) Biomolecular Sequence Analysis, v1.0. Adopted Specification, OMG document formal/01-06-08.

Object Management Group (2002) Genomic Maps, v1.0, Adopted Specification, OMG document formal/02-02-01.

Object Management Group (2003-a) Gene Expression, v1.1, OMG Final Adopted Specification, OMG document formal/03-10-01.

Object Management Group (2003-b) MDA Guide Version 1.0.1, OMG document omg/2003-06-01.

Olsen, G.J., Lane, D.J., Giovannoni, S.J., Pace, N.R. and Stalh, D.A. (1986) Microbial ecology and evolution : a ribosomal RNA approach., Ann. Rev. Microbiol., 40:337-365.

Park, T., Yi, S.G., Kang, S.H., Lee, S., Lee, Y.S. and Simon, R. (2003) Evaluation of normalization methods for microarray data, BMC Bioinformatics, 4:33.

Parkinson, H., Sarkans, U., Shojatalab, M., Abeygunawardena, N., Contrino, S., Coulson, R., Farne, A., Lara, G.G., Holloway, E., Kapushesky, M. et al (2005) ArrayExpress-a public repository for microarray gene expression data at the EBI, Nucleic Acids Res., 33(Database issue):D553-555.

Pavlidis, P., Weston, J., Cai, J. and Noble, W.S. (2002) Learning gene functional classifications from multiple data types, J Comput Biol., 9:401-411.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison, Proc Natl Acad Sci USA; 85:2444-2448.

Pepper, I.L., Gentry, T.J., Newby, D.T., Roane, T.M. and Josephson, K.L. (2002) The role of cell bioaugmentation and gene bioaugmentation in the remediation of co-contaminated soils, Environ. Health Perspect., 110: 943-946.

Peyret, P., Katinka, M.D., Duprat, S., Duffieux, F., Barbe, V., Barbazanges, M., Weissenbach, J., Saurin, W. and Vivares, C.P. (2001) Sequence and analysis of chromosome I of the amitochondriate intracellular parasite Encephalitozoon cuniculi (Microspora), Genome Res., 11:198-207.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W.L., Chen, C., Zhai, Y. (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays, Nat. Genet., 20:207-211.

Quackenbush, J. (2002) Microarray data normalization and transformation, Natuture Genetics, 32:496-501.

Rahmann, S. (2003) Fast Large Scale Oligonucleotide Selection Using the Longest Common Factor Approach, Journal of Bioinformatics and Computational Biology, 1:343-361.

Raychaudhuri, S., Stuart, J.M. and Altman, R.B. (2000) Principal components analysis to summarize microarray experiments: application to sporulation time series, Pac Symp Biocomput., 455-466.

Ren, B., Robert, F., Wyrick, J.J., Aparicio, O., Jennings, E.G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T.L., Wilson, C.J., Bell, S.P. and Young, R.A.. (2000) Genome-wide location and function of DNA binding proteins, Science, 290:2306-2309.

Reymond, N., Charles, H., Duret, L., Calevro, F., Beslon, G. and Fayard, J.M. (2004) ROSO: optimizing oligonucleotide probes for microarrays, Bioinformatics, 20:271-273.

Rimour, S., Hill, D., Militon,C., Peyret, P. (2005) GoArrays: highly dynamic and efficient microarray probe design, Bioinformatics, 21:1094-1103.

Rodgers, J.D. and Bunce, N.J. (2001) Treatment methods for the remediation of nitroaromatic explosives, Wat. Res., 35:2101-2111.

Rouillard, J.M., Herbert, C.J. and Zuker, M. (2002) OligoArray: Genome-scale oligonucleotide design for microarrays, Bioinformatics, 183:486-487.

Rouillard, J.M., Zuker, M. and Gulari, E. (2003) OligoArray 2.0: Design of oligonucleotide probes for DNA microarrays using a thermodynamic approach, Nucleic Acids Res., 31:3057-3062.

Saal, L.H., Troein, C., Vallon-Christersson, J., Gruvberger, S., Borg, A. and Peterson, C. (2002) BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data, Genome Biol., 3: software0003.1-0003.6.

Sankoff, D. and Kruskal, J. B. (1983) Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley.

SantaLucia, J.Jr. (1998) A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics, Proc Natl Acad Sci USA, 95:1460-1465.

Scherf, U., Ross, D.T., Waltham, M., Smith, L.H., Lee, J.K., Tanabe, L., et al. (2000) A gene expression database for the molecular pharmacology of cancer, Nat. Genet., 24:236-244.

Schroder, S., Weber, J. and Paul, H. (2001) 50 Nucleotide long probes on microarrays enable high signal intensity and high specificity, Genomic Discovery, MWGAG Biotech: AN014.

Segal, E., Friedman, N., Kaminski, N., Regev, A. and Koller, D. (2005) From signatures to models: understanding cancer using microarrays, Nat Genet. 37 Suppl:S38-45.

Sims, O. (2004) Enterprise MDA or How Enterprise Systems Will Be Built, MDA Journal, September 2004.

Slonim, D.K. (2002) From patterns to pathways: gene expression data analysis comes of age, Nat Genet., 32 Suppl:502-508.

Snijders, A.M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, A.K., Huey, B. and Kimura, K. (2001) Assembly of microarrays for genome-wide measurement of DNA copy number, Nat. Genet., 29:263-264.

Solinas-Toldo, S., Lampel, S., Stilgenbauer, S., Nickolenko, J., Benner, A., Dohner, H., Cremer, T., and Lichter, P. (1997) Matrixbased comparative genomic hybridization: biochips to screen for genomic imbalances, Genes Chromosomes Cancer, 20:399-407.

Sorlie, T., Perou, C.M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M.B., van de Rijn, M., Jeffrey, S.S., Thorsen, T., Quist, H., Matese, J.C., Brown, P.O., Botstein, D., Eystein Lonning, P. and

Borresen-Dale, A.L. (2001) Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications, Proc Natl Acad Sci USA, 98:10869-10874.

Southern, E.M.. (1975) Detection of specific sequences among DNA fragments separated by gel electrophoresis, J Mol Biol. 98:503-17.

Spellman, P.T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M. et al (2002) Design and implementation of microarray gene expression markup language (MAGE-ML), Genome Biol., 3:research0046.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, Mol Biol Cell., 9:3273-3297.

Stoeckert, C.J. Jr and Parkinson, H. (2003) The MGED ontology: a framework for describing functional genomics experiments, Comparative and Functional Genomics, 4:127-132.

Stoeckert, C.J. Jr, Causton, H.C. and Ball, C.A. (2002) Microarray databases: standards and ontologies, Nat Genet., 32 Suppl:469-73.

Talla, E., Tekaia, F., Brino, L. and Dujon, B. (2003) A novel design of whole-genome microarray probes for Saccharomyces cerevisiae which minimizes cross-hybridization, BMC Genomics, 4:38.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S. and Golub, T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc Natl Acad Sci USA, 96:2907-2912.

Tebbutt, S.J., He, J.Q., Burkett, K.M., Ruan, J., Opushnyev, I.V., Tripp, B.W., Zeznik, J.A., Abara, C.O., Nelson, C.C. and Walley, K.R. (2004) Microarray genotyping resource to determine population stratification in genetic association studies of complex disease, Biotechniques, 37:977-985.

Thompson, J.D., Higgins, D.G., Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Res, 22:4673-80.

Tolstrup, N., Nielsen, P.S., Kolberg, J.G., Frankel, A.M., Vissing, H. and Kauppinen S. (2003) OligoDesign: Optimal design of LNA (locked nucleic acid) oligonucleotide capture probes for gene expression profiling, Nucleic Acids Res., 31:3758-3762.

Tomiuk, S. and Hofmann, K. (2001) Microarray probe selection strategies, Briefings in Bioinformatics, 2:329-340.

Toronen, P., Kolehmainen, M., Wong, G. and Castren E. (1999) Analysis of gene expression data using self-organizing maps, FEBS Lett., 451:142-146.

Trifonov, E.N. (1990) Making sense of the human genome. In Sarma, R.H. and Sarma, M.H. (Eds), Structure & Methods Adenine Press, Albany, 1:69-77.

Troyanskaya, O.G., Arbell, O., Koren, Y., Landau, G.M. and Bolshoy, A. (2002) Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity, Bioinformatics, 18:679-688.

Tseng, G.C., Oh, M.K., Rohlin, L., Liao, J.C. and Wong, W.H. (2001), cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects, Nucleic Acids Res., 29:2549-2557.

Ukkonen, E. (1985) Finding approximate patterns in strings, J. Algo., 6:132-137.

Ukkonen, E. (1995) On-line construction of suffix-trees, Algorithmica, 14:249-260.

Valinsky, L., Della Vedova, G., Scupham, A.J., Alvey, S., Figueroa, A., Yin, B., Hartin, R.J., Chrobak, M., Crowley, D.E., Jiang, T. and Borneman, J. (2002) Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes, Appl Environ Microbiol., 68:3243-3250.

Van Gool, T., Biderre, C., Delbac, F., Wentink-Bonnema, E., Peek, R. and Vivares, C.P. (2004) Serodiagnostic studies in an immunocompetent individual infected with Encephalitozoon cuniculi, J Infect Dis., 189: 2243-2249.

van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R.

and Friend, S.H. (2002) Gene expression profiling predicts clinical outcome of breast cancer, Nature, 415:530-536.

Vivares, C.P. and Metenier, G. (2001) The microsporidian Encephalitozoon, Bioessays, 23:194-202.

Wallace, R.B., Shaffer, J., Murphy, R.F., Bonner, J., Hirose, T. and Itakura, K. (1979) Hybridization of synthetic oligodeoxyribonucleotides to phi chi 174 DNA: the effect of single base pair mismatch, Nucleic Acids Res., 6:3543-3557.

Wang, X. and Seed, B. (2003) Selection of Oligonucleotide Probes for Protein Coding Sequences, Bioinformatics, 19:796-802.

Ward, D.M., Bateson, M.M., Weller, R. and Ruff-Roberts, A.L. (1992) Ribosomal analysis of microorganisms as they occur in nature, Adv. Microbial Ecology, 12:219-286.

Webb, S.C., Attwood, A., Brooks, T., Freeman, T., Gardner, P., Pritchard, C., Williams, D., Underhill, P., Strivens, M.A., Greenfield, A. and Pilicheva, E. (2004) LIMaS: the JAVA-based application and database for microarray experiment tracking, Mammalian Genome, 15:740-747.

Weiner, P. (1973) Linear pattern matching algorithms. In Proc. Of the 14th IEEE Symp. on Switching and Automata Theory, 1-11.

Weiss, L.M. (2001) Microsporidia: emerging pathogenic protests, Acta Trop., 78: 89-102.

Wetmur, J.G. (1991) Applications of the principles of nucleic acid hybridization. Crit. Rev. Biochem. Mol. Biol., 26:227-259.

Wheeler, D.L., Chappey, C., Lash, A.E., Leipe, D.D., Madden, T.L., Schuler, G.D., Tatusova, T.A., Rapp, B.A. (2000) Database resources of the National Center for Biotechnology Information, Nucleic Acids Res, 28:10-4.

Wheeler, D.L., Church, D.M., Edgar, R., Federhen, S., Helmberg, W., Madden, T.L., Pontius, J.U., Schuler, G.D., Schriml, L.M., Sequeira, E. et al (2004) Database resources of the National Center for Biotechnology Information: update, Nucleic Acids Res., 32(Database issue):D35-40.

Widada, J., Nojiri, H. and Omori, T. (2002) Recent developments in molecular techniques for identification and monitoring of xenobiotic-degrading bacteria and their catabolic genes in bioremediation, Appl. Microbiol. Biotechnol., 60:45-59.

Williams, A.L. and Tinoco, I.Jr. (1986) A dynamic programming algorithm for finding alternate RNA secondary structures, Nucleic Acids Res., 14:299-315.

Wilson, K.H., Wilson, W.J., Radosevich, J.L., DeSantis, T.Z., Viswanathan, V.S., Kuczmarski, T.A. and Andersen, G.L. (2002) High-density microarray of small-subunit ribosomal DNA probes, Appl Environ Microbiol., 68:2535-2541.

Woese, C.R., Fox, G.E., Zablen, L., Uchida, T., Bonen, L., Pechman, K., Lewis, B.J. and Stahl, D. (1975) Conservation of primary structure in 16S ribosomal RNA, Nature, 254:83-86.

Wolfinger, R.D., Gibson, G., Wolfinger, E.D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C. and Paules, R.S. (2001) Assessing gene significance from cDNA microarray expression data via mixed models, Journal of Computational Biology, 8(6):625-637.

Woolf, P.J. and Wang, Y. (2000) A fuzzy logic approach to analyzing gene expression data, Physiol Genomics, 3:9-15.

Wootton, J.C. and Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases, Methods Enzymol., 266:554-571.

Wu, S. and Manber, U. (1992) A fast text searching allowing errors, Commun. ACM, 35:83-91.

Yang, I.V., Chen, E., Hasseman, J.P., Liang, W., Frank, B.C., Wang, S., Sharov, V., Saeed, A.I., White, J., Li, J., Lee, N.H., Yeatman, T.J., Quackenbush, J. (2002) Within the fold: assessing differential expression measures and reproducibility in microarray assays, Genome Biol., 3:research0062.

Yang, Y.H., Dudoit, S., Luu, P., Lin, D.M., Peng, V., Ngai, J., Speed, T.P. (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation, Nucleic Acids Res., 30:e15.

Yeung, K.Y. and Ruzzo, W.L. (2001) Principal component analysis for clustering gene expression data, Bioinformatics, 17:763-774.

Zhang, Z., Willson, R.C. and Fox, G.E. (2002) Identification of characteristic oligonucleotides in the bacterial 16S ribosomal RNA sequence dataset, Bioinformatics, 18:244-250.

Zuker, M. (2000) Calculating nucleic acid secondary structure, Curr. Opin. Struct. Biol., 10:303-310.

Zuker, M., Mathews, D.H. and Turner, D.H. (1999) Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide. In RNA Biochemistry and Biotechnology. Edited by Barciszewski J, Clark BFC. Dordrecht: Kluwer Academic Publishers, 11-43.