Bioinformatique des ARNs non-codants : Algorithmes pour leur identification et la prédiction de leur structure

Mémoire présenté par

Fariza TAHI

en vue de l'obtention du diplôme d'

Habilitation à Diriger des Recherches

Université d'Evry-Val d'Essonne

soutenue le 17 juillet 2014 devant le jury composé de :

Dr. Emmanuel Barillot	(Institut Curie)	Examinateur
Prof. Mokrane Bouzeghoub	(UVSQ)	Examinateur
Prof. Peter Clote	(Boston College)	Rapporteur
Prof. Florence d'Alché	(UEVE)	Examinateur
Dr. Christine Gaspin	(INRA Toulouse)	Rapporteur
Prof. Daniel Gautheret	(Université Paris-Sud)	Examinateur
Prof. Christian Michel	(Université de Strasbourg)	Rapporteur
Dr. Mireille Régnier	(INRIA/LIX)	Examinateur

A Clara ...

Table des matières

1 Introduction **1**

2 Le monde des ARNs non-codants **4**

2.1 La biologie structurale et les ARNs non-codants 4

2.1.1 La structure secondaire des ARNs non-codants 4

2.1.2 Les pseudonœuds 6

2.2 Quelques exemples d'ARNs non-codants 7

2.2.1 L'ARN de transfert 7

2.2.2 L'ARN ribosomique 7

2.2.3 Les microARNs 7

2.2.4 Les piARNs (ARNs interagissant avec les protéines PIWI) 9

2.2.5 Autres exemples d'ARNs non-codants 10

2.3 Les éléments transposables et les petits ARNs non-codants 10

2.3.1 Les éléments transposables 10

2.3.2 Les ARNs non-codants dérivés des éléments transposables 11

2.3.3 Les piARNs, moyen de défense contre les éléments transposables 11

3 Prédiction de structures secondaires d'ARN **13**

3.1 Introduction 13

3.1.1 Approches principales existantes 13

3.1.2 Problème de recherche de pseudonœuds 14

3.1.3 Notre contribution 15

3.2 Définitions 17

3.2.1 Définitions et représentations 17

3.2.2 Mesures utilisées pour l'évaluation des résultats de prédiction 19

3.3 *DCfold* : Prédiction de structures secondaires d'ARN basée sur l'approche comparative . . . 20

3.3.1 Notre approche 20

3.3.2 Critères de sélection des hélices 20

3.3.3 Recherche des hélices conservées . . . 21
3.3.4 Approche “diviser pour régner” . . . 22
3.3.5 Algorithme . . . 23
3.3.6 Conclusion . . . 25
3.4 *P-DCfold* : Prédiction de pseudonœuds . . . 26
3.4.1 Notre approche . . . 26
3.4.2 Algorithme . . . 26
3.4.3 Résultats . . . 27
3.4.4 Conclusion . . . 29
3.5 *SSCA* : Sélection de séquences homologues pour l'approche comparative . . . 30
3.5.1 Pourquoi sélectionner les séquences homologues . . . 30
3.5.2 Critères pour la sélection des séquences homologues . . . 31
3.5.3 Algorithme . . . 33
3.5.4 Résultats . . . 35
3.5.5 Conclusion . . . 36
3.6 Tfold : Algorithme efficace pour la prédiction de structure secondaire d'ARN incluant les pseudonœuds . . . 37
3.6.1 Principe et Algorithme . . . 37
3.6.2 Résultats . . . 41
3.6.3 Conclusion . . . 44

4 Prédiction de précurseurs de microARNs 46
4.1 Introduction . . . 46
4.1.1 Approches existantes et état de l'art . . . 46
4.1.2 Problématiques et enjeux . . . 47
4.1.3 Notre contribution . . . 47
4.2 *miRNAFold* : Recherche *ab-initio* de précurseurs de microARNs dans les génomes . . . 49
4.2.1 Caractéristiques des précurseurs de miARNs . . . 49
4.2.2 Notre approche et algorithme pour l'identification des miARNs . . . 51
4.2.3 Résultats . . . 51
4.2.4 Passage à l'échelle : version GPU de l'algorithme *miRNAFold* . . . 54
4.2.5 Conclusion . . . 56
4.3 miRBoost : Classification des vrais et faux précurseurs de microARNs . . . 57
4.3.1 Problème des données déséquilibrées . . . 57
4.3.2 Méthode de boosting . . . 57
4.3.3 Notre approche . . . 58

4.3.4 Sélection des caractéristiques utilisées 59
4.3.5 Résultats 59
4.3.6 Conclusion 62

5 Prédiction de petits ARNs non-codants en lien avec les éléments transposables 63

5.1 Introduction 63
5.1.1 Problème d'annotation des ARNncs liés aux éléments transposables 63
5.1.2 Les piARNs, petits ARNncs dérivés des éléments transposables et encore mal connus 64
5.1.3 Notre contribution 65
5.2 *ncRNAclassifier* : Identification des ARNs non-codants dérivés d'éléments transposables . . 66
5.2.1 Notre approche 66
5.2.2 Description de la méthode 66
5.2.3 Résultats 67
5.2.4 Conclusion 70
5.3 *piRPred* : Prédiction de piARNs 70
5.3.1 Problématique de l'identification des piARNs et approche développée 70
5.3.2 Description des noyaux développés 71
5.3.3 Résultats 72
5.3.4 Conclusion 74

6 Plateforme logicielle EvryRNA 76

7 Travaux de collaboration en cours 79

7.1 Recherche de biomarqueurs ARNncs de la Dystrophie Musculaire de Duchenne 79
7.2 Détermination d'ARNncs impliqués dans la différenciation sexuelle chez les plantes 80

Chapitre 1

Introduction

Mes travaux de recherche s'inscrivent dans le domaine de la "Bioinformatique". Je m'intéresse au développement de méthodes informatiques et bioinformatiques pour la compréhension du vivant. Mes contributions concernent principalement la prédiction et la recherche d'ARNs non-codants. Je me suis par ailleurs intéressée aux problématiques d'intégration de données et de modèles biologiques.

Mes travaux, en particulier ceux concernant les ARNs non-codants, sont motivés et guidés par l'application. En d'autre termes, mon objectif premier est de proposer des outils "efficaces" pour résoudre des problématiques biologique/bioinformatiques données et essayer ainsi de participer, à mon échelle, à cette quête de la compréhension du vivant.

J'ai découvert la bioinformatique lors de ma thèse de doctorat, que j'ai effectuée à l'INRIA Rocquencourt, dans l'Action transversale Genome. A cette période, c'est-à-dire les années 90, la bioinformatique était essentiellement centrée sur l'analyse de séquences. Mes travaux ont ainsi porté sur le développement de méthodes formelles pour l'analyse des séquences génomiques. Je m'étais intéressée à la recherche de répétitions dans les séquences d'ADN et d'ARN, à la compression de séquences d'ADN et à la prédiction de structures secondaires d'ARN, principalement via des méthodes d'algorithmique de séquences, ainsi qu'à l'énumération de ces structures par des séries génératrices.

Après ma thèse, j'ai voulu découvrir d'autres aspects de la bioinformatique, et pourquoi pas dans une équipe de biologie. J'ai donc intégré (pour un post-doctorat de 2 ans) l'équipe Genexpress dirigée par Charles Auffray au CNRS, équipe biologique avec une composante bioinformatique, qui travaillait notamment sur les puces à ADN. Les technologies des puces à ADN (initialement macroarrays puis microarrays) étaient en plein essor, et soulevaient d'importantes problématiques d'organisation et d'analyse de données. J'ai ainsi travaillé sur cette problématique, qui devenait à ce moment là l'un des thèmes de recherche les plus importants en bioinformatique. Au sein de cette équipe, j'ai également découvert la problématique d'intégration de données, un sujet qui démarrait à peine dans le monde de la bioinformatique. Dans l'équipe on avait développé une base de connaissances où on intégrait les données d'expression générées dans l'équipe avec des données issues de sources de données externes, telles que des données de séquences et des données de cartographie. Lorsque j'ai intégré mon poste de maître de conférences à l'Université d'Evry-Val d'Essonne, l'intégration de données biologiques devenait un thème de recherche très important en bioinformatique. J'ai donc tout naturellement continué à travailler sur cette problématique.

Avec les données d'expression générées via les puces à ADN, on construisait de plus en plus de réseaux de régulation, qu'il fallait étudier, en l'occurrence étudier leur dynamique, leur états stables, etc. Ainsi, à l'ère de la biologie systémique, et profitant du contexte Evryen, je me suis également intéressée au thème de recherche qu'est la modélisation et l'intégration des réseaux de régulation.

Par ailleurs, de plus en plus de travaux montraient l'importance des ARNs non-codants et leurs implication dans de nombreux processus de régulation, et la bioinformatique allait jouer un rôle de plus en plus important

dans leur prédiction et leur identification. Ayant déjà abordé pendant ma thèse de doctorat des travaux liés à la bioinformatique des ARNs non-codants, je me suis ré-intéressée à ce thème et je continue à m'y intéresser, tant c'est un domaine fascinant. Le monde des ARNs est en effet loin d'avoir délivré tous ses secrets et n'arrête de surprendre, soulevant ainsi à chaque découverte de nouveaux défis bioinformatiques.

Dans ce rapport, je présente mes travaux et contributions en Bioinformatique des ARNs non-codants, domaine dans lequel je me suis le plus investit ces dernières années. Les autres travaux sont décrits brièvement dans mon curriculum vitae détaillé.

Dans le domaine des ARNs non-codants, j'ai mené différents travaux. Je me suis d'abord intéressée à la problématique de prédiction des structures secondaires d'ARNs. Ces travaux, effectués en grande partie dans le cadre de la thèse de Stéfan Engelen, ont permis de proposer plusieurs algorithmes : DCfold, un algorithme pour la prédiction de structure secondaire d'ARN basé sur l'approche comparative ; P-DCfold, une extension de DCfold pour la prédiction de pseudonœuds ; SSCA, un algorithme pour la sélection des séquences homologues à utiliser dans un algorithme de prédiction de structure secondaire basé sur l'approche comparative ; et enfin Tfold, qui intègre SSCA et *P-DCfold* pour offrir une méthode complète et efficace permettant de prédire la structure secondaire incluant les pseudonœuds d'une séquence d'ARN donnée, en prenant en entrée un ensemble "brut" de séquences homologues alignées. Ces algorithmes sont décrits en Chapitre 3.

Je me suis ensuite intéressée à l'identification dans des séquences génomiques d'un type particulier d'ARNs non-codants, les microARNs. Les microARNs (miARNs) sont des petits ARNs, de 19 à 25 nt, dont les précurseurs, de taille entre 60 et 300 nt, ont une structure particulière en "hairpin" ou "épingle à cheveux". Nous avons développé, avec le post-doctorant Sébastien Tempel, un algorithme, miRNAFold, qui permet de rechercher les structures en hairpin des précurseurs de miARNs dans des génomes entiers. Puis avec le post-doctorant Van Du Tran et en collaboration avec Farida Zehraoui de l'équipe AROBAS d'IBISC, nous avons développé un second algorithme, miRBoost, basé sur de l'apprentissage automatique, qui permet de classifier une séquence en pré-miARN ou non pré-miARN. Ces deux algorithmes sont décrits en Chapitre 4.

Lors de ce travail sur les miARNs, nous nous sommes rendus compte que certains miARNs de miRBase (la base de données de référence des miARNs répertoriés) semblaient correspondre à des éléments transposables (ET). Les éléments transposables (ou transposons) sont des séquences capables de se déplacer et se dupliquer dans le génome de manière autonome. Plusieurs travaux montrent que certains miARNs dérivent d'éléments transposables, et que certains sont même mal-annotés car correspondant à des éléments transposables. D'autres ARNs non-codants dérivent d'ETs, tels que les snoARNs (petits ARNs nucléaires) ou les piARNs (ARNs interagissant avec les protéines PIWI). Avec Sébastien Tempel et en collaboration avec Nicolas Pollet de iSSB, nous avons développé un outil automatique, appelé ncRNAclassifier, permettant de déterminer si un petit ARNnc est dérivé d'un ET, ou éventuellement correspond complètement à un ET et est donc mal-annoté. Ce travail est décrit en Chapitre 5.

Toujours en lien avec les éléments transposables, je me suis récemment intéressée à la prédiction des piARNs, dans le cadre du stage de Jocelyn Brayet co-encadré avec Farida Zehraoui et David Israeli du Généthon. Les piARNs sont des petits ARNs entre 20 et 31 nt, qui ont pour rôle principal de protéger le génome contre l'invasion d'ETs. Ce sont des ARNncs très récemment découverts et encore très mal connus. Contrairement aux miARNs, ils sont très peu conservés entre espèces et on ne leur connaît pas de structure particulière, rendant leur prédiction par des méthodes automatiques très difficile. Nous avons proposé un algorithme extensible et adaptative, basé sur de l'apprentissage automatique permettant la classification d'une séquence donnée en piARN ou non-piARN. Ce travail, qui a donné des résultats tout à fait prometteurs, est également décrit en Chapitre 5.

Afin de mettre à disposition de la communauté scientifique tous nos outils bioinformatiques destinés à la prédiction et l'identification des ARNs non-codants, nous avons développé un serveur web, une plateforme logicielle appelée EvryRNA disponible à l'adresse `http://EvryRNA.ibisc.univ-evry.fr`. Cette plateforme est présentée brièvement en Chapitre 6.

Enfin, en Chapitre 7 nous présentons les travaux de collaboration que nous menons actuellement avec des biologistes autour de l'identification d'ARNncs. Nous collaborons d'une part avec le Généthon autour de l'identification d'ARNncs comme possibles biomarqueurs de la Dystrophie Musculaire de Duchenne et d'autre part avec l'URGV autour de l'identification d'ARNncs impliqués dans la différenciation sexuelle chez les plantes.

Chapitre 2

Le monde des ARNs non-codants

2.1 La biologie structurale et les ARNs non-codants

La connaissance de la structure des macromolécules, de leurs interactions et associations est au cœur de la compréhension du fonctionnement du vivant. Pendant longtemps, lorsque l'on parlait de biologie structurale, on pensait tout d'abord à la structure des protéines, et on oubliait le rôle important dans ce domaine des ARNs non codants, qui adoptent aussi une structure tridimensionnelle (Figure 2.1), et qui sont impliqués via leur interactions structurelles dans de nombreux processus biologiques.

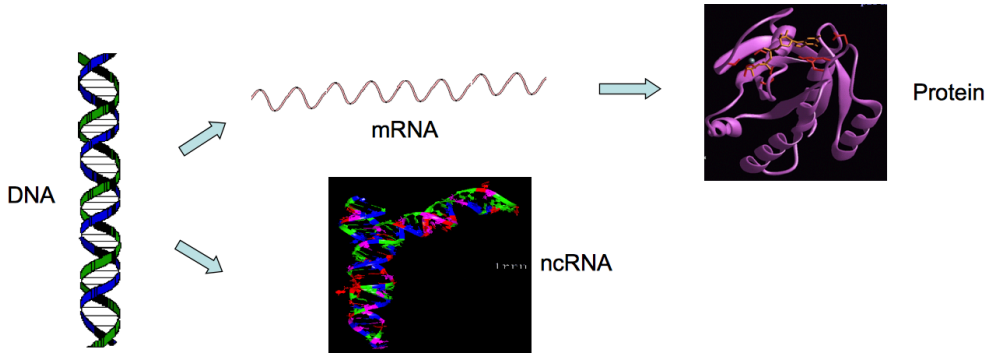

FIGURE 2.1 – Les fonctions biologiques sont exprimées à travers deux voies : les protéines et les ARNs. Le protéines sont codées à partir de l'ARN messager. Les ARNs non-codants ont leur propre structure et ne codent pas pour des protéines.

Dans les années 80, la découverte de l'ARN catalytique a révolutionné les perceptions des origines de la vie et a conduit au développement de nombreuses recherches sur les ARNs. Ainsi, aujourd'hui, on ne cesse d'accumuler des connaissances sur les structures d'ARN, de leurs processus de repliement, d'évolution et de catalyse. Les ARNs sont maintenant au cœur de nombreux travaux de recherche tant leurs rôles s'avèrent très importants dans un nombre considérable de processus biologiques. La découverte des petits ARNs interférants, intervenant dans l'inhibition de la régulation, a ouvert un champ d'investigation considérable, aussi bien en biologie fondamentale qu'en médecine.

2.1.1 La structure secondaire des ARNs non-codants

Les molécules d'ARN présentent trois niveaux de structuration (voir Figure 2.2) :

– La structure primaire, qui correspond à la séquence - linéaire -, composée d'une suite de nucléotides :

Adénine (A), Cytosine (C), Guanine (G) et Uracile (U).

– La structure secondaire, qui correspond au repliement de la séquence via des appariements de bases de Watson-Crick (A-U et G-C) et de Wooble (G-U). Ces appariements sont formés par des liaisons hydrogène entre les nucléotides correspondants.
– La structure tertiaire, qui est la conformation tridimensionnelle de la chaîne de nucléotides. Elle résulte du repliement de la structure secondaire via des appariements non canoniques (autres que Watson-Crick). Ces appariements sont divers, plus de 150 types ont été observés [111].

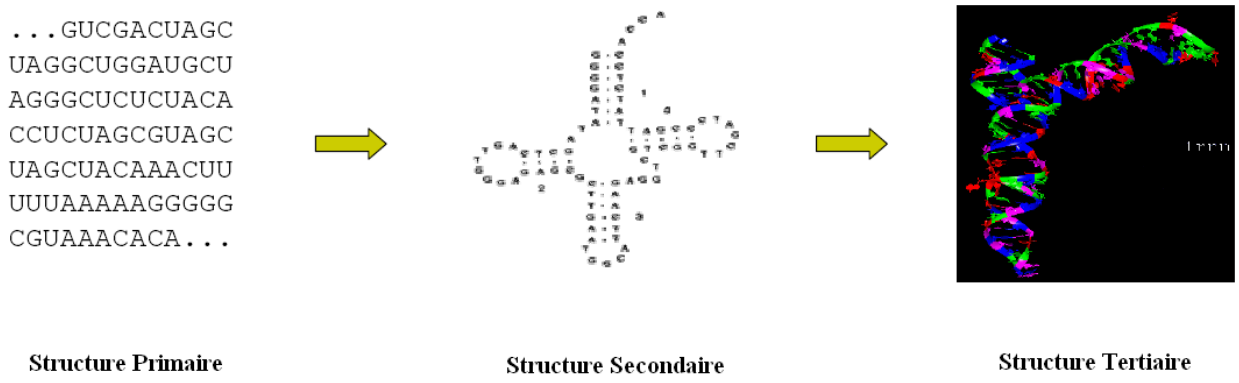

FIGURE 2.2 – Les trois niveaux de structure des ARNs non-codants.

La structure secondaire d'un ARN correspond à la forme (ou topologie) induite par l'ensemble des appariements A-U, G-C et G-U de la molécule simple brin. Elle est ainsi composée par des régions appariées, appelées hélices (ou tiges), et des régions non-appariées, appelées boucles. Il y a différents types de boucles (voir Figure 2.3) :

– les boucles internes, qui relient deux hélices, peuvent être soit symétriques (les deux brins composant la boucle sont de tailles égales), soit non-symétriques ;
– les renflements, reliant deux hélices par un seul brin non-apparié, les deux hélices étant côte à côte sur l'autre brin ;
– les boucles terminales, situées à l'extrémité d'un "bras" de la structure ; elles ont une taille minimale de quatre nucléotides, correspondant à la distance minimale nécessaire entre deux nucléotides de la molécule pour s'apparier ;
– les boucles multiples, qui relient plusieurs hélices (trois ou plus).

On retrouve ainsi plusieurs types de motifs dans la structure secondaire. Le plus connu est le motif de "tiges boucle" ou "épingle à cheveux", constitué d'une hélice, éventuellement une succession d'hélices séparées par des renflements ou des boucles internes (hélices non exactes), et d'une boucle terminale (voir Figure 2.4). Plusieurs petits ARNs tels que les microARNs présentent cette forme (voir Section 2.2.3).

Une structure secondaire est donc essentiellement définie par sa forme, et plus exactement par la position et la longueur de ses hélices. Mais la nature des bases non appariées n'est pas toujours sans importance. Elles peuvent intervenir dans la formation de la structure tertiaire, en s'appariant entre elles (via des interactions autres que celles de Watson-Crick et de Wooble) en entraînant ainsi un enroulement de la structure secondaire. Elles peuvent également jouer un rôle dans la fonction biologique qui leur est associée (par exemple, dans la boucle terminale de la feuille de trèfle de l'ARN de transfert, les trois bases formant l'*anticodon* permettent la transcription d'un codon en un acide aminé (voir Section 2.2.1)).

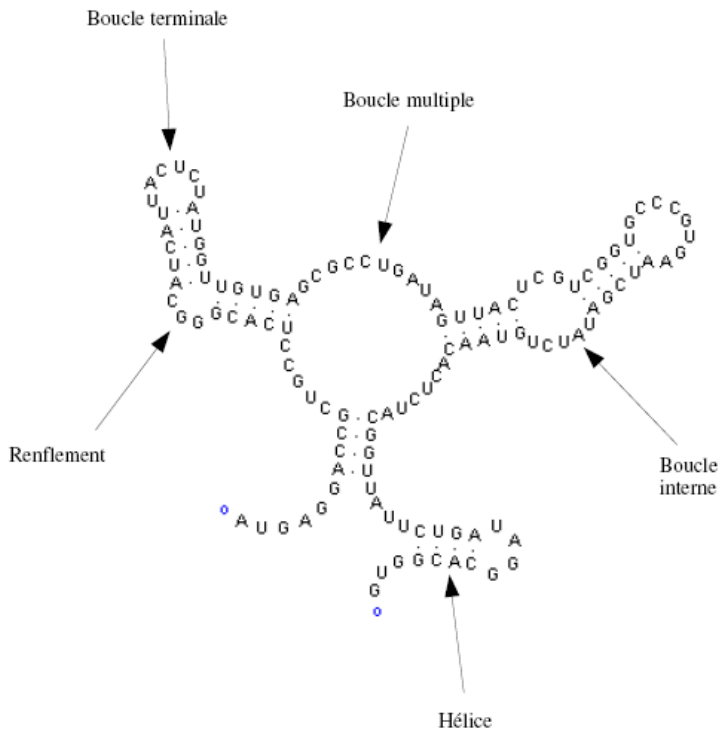

FIGURE 2.3 – Composants des structures secondaires d'ARNs non-codants.

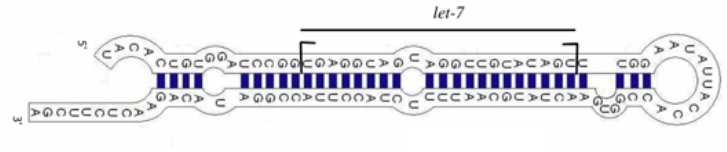

FIGURE 2.4 – Exemple de structure en épingle à cheveux (hairpin).

2.1.2 Les pseudonœuds

De nombreuses molécules d'ARN contiennent des pseudonœuds. Un pseudonœud est une structure formée par l'appariement d'une boucle avec une région de l'ARN située à l'extérieur de l'hélice qui la délimite (Figure 2.5).

Les pseudonœuds sont généralement exclus de la définition classique de la structure secondaire. Une structure secondaire d'ARN sans pseudonœuds est une structure plane (en deux dimensions). Lorsqu'elle contient des pseudonœuds, elle perd la conformation plane. Pour cette raison, les pseudonœuds sont souvent considérés comme faisant partie de la structure tertiaire. Or les liaisons intervenant dans les pseudonœuds sont de la même nature que ceux de la structure secondaire, c'est à dire de simples appariements A-U, C-G et G-U.

De plus en plus d'études attribuent un rôle non négligeable aux pseudonœuds, en particulier dans la régulation de certains processus biologiques. Des observations expérimentales ont suggéré des rôles de "commutateurs" ou d'éléments de contrôle dans plusieurs fonctions biologiques [167]. Dans les molécules qui n'ont pas une conformation tridimensionnelle globale, les pseudonœuds permettent un repliement au niveau local et leurs positions le long de la séquence reflètent alors leurs fonctions [121].

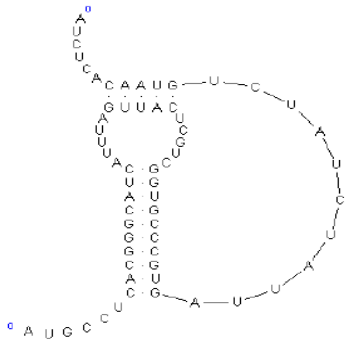

FIGURE 2.5 – Exemple de pseudonœud.

2.2 Quelques exemples d'ARNs non-codants

2.2.1 L'ARN de transfert

L'ARN de transfert (ARNt) est un ARN composé de 73 à 93 nucléotides présents dans tous les organismes vivants. Il est utilisé dans la traduction de l'ARN messager (ARNm) en protéine. Il est caractérisé par une structure secondaire en feuille de trèfle composée de trois hélices (la Figure 2.2 présente un exemple d'ARNt). De par sa forme, cette structure joue un rôle fondamental dans la transcription de chaque triplet de nucléotides en acide aminé. Via la boucle terminale de l'une de ses hélices (cette boucle est appelée anticodon), elle permet d'associer à chaque codon de l'ARNm un acide aminé [95, 33], puis de fournir celui-ci au complexe ribosomique qui va allonger la chaîne protéique.

2.2.2 L'ARN ribosomique

Un autre ARN largement connu et faisant partie des premiers ARNs découverts est l'ARN ribosomique (ARNr). L'ARNr est le composant principal du ribosome dans tous les organismes vivants. Le ribosome contient deux sous-unités, une petite et une grande. La petite sous-unité est composée de l'ARNr 16S dans les organismes procaryotes et l'ARNr 18S dans les organismes eucaryotes. Son rôle est de lire l'ARNm et de vérifier la compatibilité entre le codon de l'ARNm et l'anticodon de l'ARNt. La grande sous-unité est composée de l'ARNr 5S et l'ARNr 23S chez les procaryotes, et de l'ARNr 5S et l'ARNr 28S ou 25S chez les eucaryotes (28S chez les animaux et 25S chez les plantes). Elle permet la création de la liaison peptidique entre les acides aminés [138].

Les différents ARNs ribosomiques ont des tailles très variables. Le plus grand est l'ARNr 28S, composé de 4 800 nucléotides. L'ARNr 23S est composé de 2 300 nucléotides, l'ARNr 18S de 1 900 nucléotide, l'ARNr 16S de 1 500 nucléotides, et le plus petit d'entre eux, l'ARNr 5S a environ 120 nucléotides (voir Figure 2.6).

2.2.3 Les microARNs

Les microARNs (miARNs) sont des petits ARNs non-codants de longueur de séquence de seulement 21-25 nt [9, 75]. Ils sont impliqués en tant que régulateurs négatifs de l'expression génique au niveau post-transcriptionnel, en se liant à des objectifs spécifiques d'ARNm dont les traductions sont inhibées ou régulées à la baisse [75, 109].

Selon la compréhension actuelle de la biogenèse des miARNs, les gènes de miARN sont d'abord transcrits en de longs miARNs primaires (pri-miARNs), puis sont clivés en des longs précurseurs de miARNs (pré-

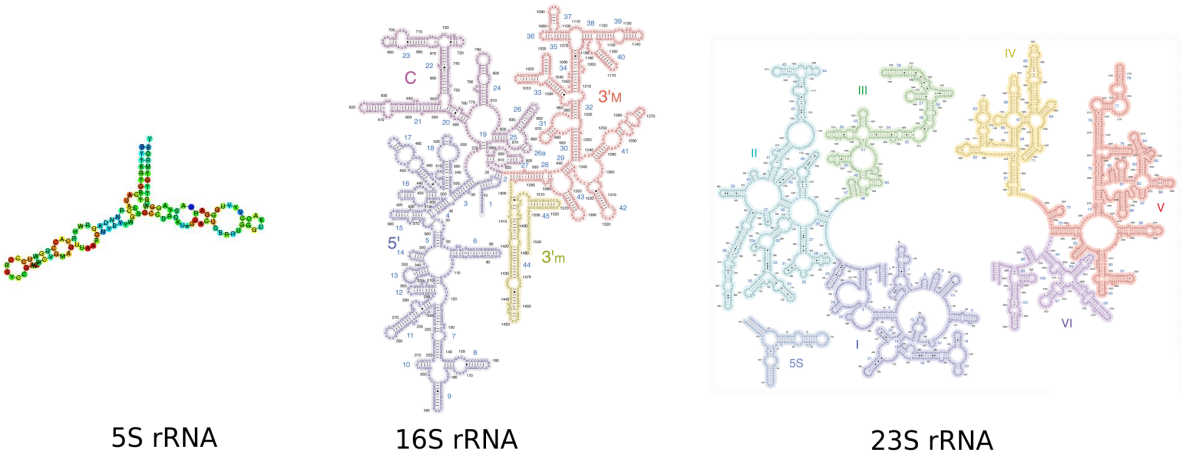

FIGURE 2.6 – Exemples de structures secondaires d'ARNs ribosomiques : 5S, 16S et 23S.

miARNs), de longueurs entre 60 et 140 nt [98], par le complexe Drosha/Pacha. Le pré-miARN, qui a une structure en épingle à cheveux, est transporté dans le cytoplasme par Exportin-5 et clivé par Dicer en un miARN mature [9]. Dans le complexe RISC, un miARN se lie à un transcrit d'ARNm spécifique et conduit à la coupure ou la dégradation de l'ARNm (voir Figure 2.7).

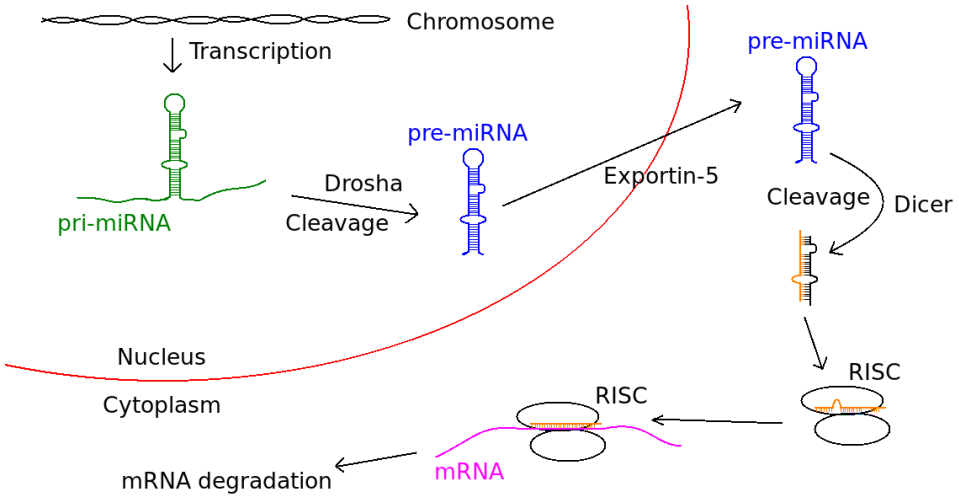

FIGURE 2.7 – Biogenèse des miARNs.

Les dé-régulations provoquées par les miARNs sont à l'origine d'un grand nombre de maladies (répertoriées dans la base de données miR2Disease [88])), telles que le cancer [76, 134], les maladies neuro-dégénératives comme l'Alzheimer [120], les maladies cardiaques [188], etc. L'identification de miARNs est donc très important aussi bien pour les sciences biologiques que médicales.

Plus de 18 000 miARNs ont été découverts dans environ 140 espèces, dont plus de 1 500 chez l'Homme [98]. Et de récentes études ont révélé qu'un large nombre de miARNs n'ont pas encore été découverts [195]. Tous les miARNs découverts (et publiés) sont répertoriés dans la base de données miRBase [98, 130].

2.2.4 Les piARNs (ARNs interagissant avec les protéines PIWI)

Les piARNs, ou ARNs interagissant avec les protéines PIWI, sont des petits ARNs non codants simple brin, de 24 à 35 nucléotides [113], découverts récemment et encore peu caractérisés, contrairement aux miARNs. Ils ont été découverts dans les cellules de la lignée germinale, jouant un rôle important dans la protection du génome contre l'invasion d'éléments transposables [40, 100, 22] (voir Section 2.3). En plus de leur activité dans les cellules germinales, l'accumulation de données suggèrent que les piARNs sont également présents dans les cellules somatiques, et seraient même impliqués dans des maladies telles que le cancer [128]. Une nouvelle vision propose ainsi une définition plus large de l'expression et de la fonction biologique des piARNs à la fois dans la lignée germinale et dans les cellules somatiques [147, 162].

Les piARNs représentent la plus large et la plus hétérogène des classes de petits ARNncs, dépassant par exemple les 2 millions de piARNs distincts chez la souris [108]. Ils sont non conservés entre espèces et également au sein d'une même espèce, contrairement aux miARNs par exemple. A ce jour, on ne leur connaît pas de motifs de structure secondaire, ni de motifs communs au séquences, à part la présence d'un uridine (U) en première base en 5' [113]. Les deux caractéristiques adoptées et reconnues sont leurs longueurs variant entre 24 et 35 nt et leur présence dans des locus sous forme de clusters, allant de 1 kb à plus de 100 kb de long. Plusieurs clusters de piARNs ont été découverts chez différentes espèces. Ils sont répertoriés dans le serveur piRNABank [102, 153].

Selon les connaissances actuelles, les différentes étapes de la biogénèse d'un piARN se déroulent à proximité de la membrane nucléaire externe, nommée nuage granulé. En effet, des études ont montré que les protéines PIWI impliquées dans ce processus se trouvent dans cet endroit de la cellule. La production de piARNs matures se fait en plusieurs étapes [113] (voir Figure 2.8) :

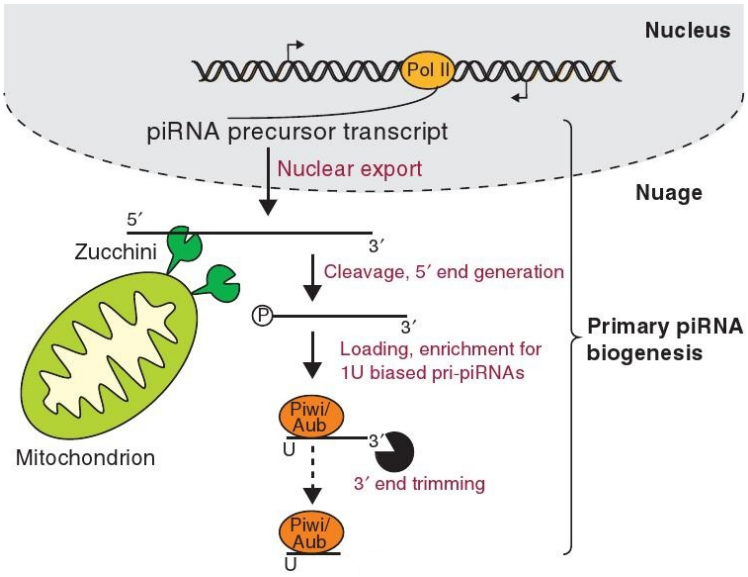

FIGURE 2.8 – Biogénèse des piARNs [113]

Il y a d'abord fixation d'un facteur de transcription (le myb-A) sur le promoteur du cluster, permettant l'obtention d'un piARN précurseur simple brin. Ce précurseur de piARN est ensuite transporté dans le cytoplasme de la cellule. L'extrémité 5' du précurseur de piARN est ensuite clivé par la protéine Zucchini chez la drosophile et MitoPLD chez la souris. Le précurseur de piARN est alors lié à une protéine de la famille des PIWI (Aubergine ou Ago3 chez la drosophile). A ce stade, les piARNs ont une forte tendance à avoir un uridine (U) en première base en 5'. L'autre extrémité (3') est ensuite coupée par un facteur non identifié appelé "tondeuse". Cette coupure en 3' s'arrête quand la tondeuse atteint la région de l'ARN protégée par l'empreinte de la protéine PIWI. On retrouve ainsi des piARNs de longueurs différentes en

fonction de la protéine PIWI qui interagit avec le précurseur du piARN. Le piARN devient mature après l'action de la protéine Hen1 qui ajoute un groupement 2'-O-méthyl en 3'.

2.2.5 Autres exemples d'ARNs non-codants

L'ARNtm L'ARNtm (également connu sous le nom d'ARN 10SA ou SsrA) joue un rôle important dans la traduction. Il combine à la fois les propriétés de l'ARNt et de l'ARNm, afin de résoudre des problèmes découlant de ribosomes bloqués dans la traduction [49]. De taille autour de 300 nucléotides, cet ARN présente des pseudonœuds. La structure secondaire de Escherichia Coli par exemple contient quatre pseudonœuds (voir Figure 3.9). La base de données tmRDB [220] fournit des séquences alignées, annotées et phylogénétiquement ordonnées d'ARNtm de plusieurs centaines d'espèces, ainsi que la structure secondaire et tridimensionnelle de certains d'entre eux.

L'ARN Ribonucléase P L'ARN Ribonucléase P ou RNase P est une enzyme présente dans toutes les cellules vivantes et dont la fonction est la maturation des ARNt [4, 37, 146]. Sa séquence est composée d'environ 380 nucléotides et sa structure secondaire contient deux pseudonœuds (voir Figure 3.9). Une compilation de séquences, d'alignements de séquences, de structures secondaires et de modèles tridimensionnels des ARN RNase P sont donnés dans la base de données P RNase [20].

L'ARN SRP L'ARN SRP (également connu sous le nom d'ARN 7SL, 6S ou 4.5S) est un composant du complexe ribonucleoprotéine de la molécule de reconnaissance de signal (SRP). La SRP permet aux protéines d'être sécrétées, et contribue à la fixation et la libération du peptide signal. Cet ARN, d'environ 300 nucléotides, a une structure secondaire composée d'une vingtaine d'hélices (voir Figure 3.9) et contenant chez certaines espèces, par exemple chez l'Halobacterium halobium, un pseudonœud. La base de Données Signal Recognition Particle Database [60] fournit des séquences d'ARN SRP ainsi que des alignement de séquences.

2.3 Les éléments transposables et les petits ARNs non-codants

2.3.1 Les éléments transposables

Les éléments transposables (ETs), appelés aussi transposons, sont des éléments fonctionnels répandus dans les génomes où ils se déplacent ou sont copiés d'une location génomique à une autre [31]. Les ETs représentent à eux seuls une fraction substantielle de nombreux génomes eucaryotes [30]. Ils constituent un moteur essentiel de l'évolution et de la diversité entre les espèces. A titre d'exemple, 45% du génome de l'Homme est composé de transposons ou de séquences dérivées de transposons.

Les éléments transposables sont caractérisés et classés sur la base de structures terminales ou sous-terminales et/ou sur leur capacité de codage en protéine [202]. Ils sont divisés en deux classes : Classe I et Classe II. Les éléments de Classe I (rétrotransposons) utilisent la transcription inverse de l'ARN à partir d'un intermédiaire et les éléments de Classe II (transposons d'ADN) sont caractérisés par des répétitions terminales inversées (TIRs) et sont mobilisés par une transposase [31].

Beaucoup de familles de ETs ne montrent pas de capacité codante de la protéine et sont appelés éléments transposables non-autonomes [31]. Par exemple, les SINEs (Short INterspersed Elements) tels que *Alu* sont des éléments non-autonomes de Classe I, caractérisés par des séquences courtes (100 à 500 nt) qui présentent des structures secondaires stables semblables à la fusion d'un ARNt et d'une structure en épingle à cheveux [93, 175]. Un autre exemple concerne les MITEs (Miniature Inverted-repeat Transposable Elements), des

éléments non-autonomes de Classe II caractérisés par une petite taille (80-500 pb) et une structure secondaire en épingle à cheveux stable [28].

2.3.2 Les ARNs non-codants dérivés des éléments transposables

Les petits ARNs fonctionnels (miARNs, snoARNs, siARNs, piARNs ...) sont produits par plusieurs voies de biosynthèse qui métabolisent des structures en épingle à cheveux formées par les ARNs précurseurs provenant de gènes répétés inversés [132, 193]. Le réservoir de ces hairpins dans les grands génomes est énorme, et une grande partie de ces génomes est transcrite (93% du génome humain) [92] et physiologiquement transformée en grands et petits morceaux d'ARNs, y compris les épingles à cheveux [144]. Il s'avère que la majorité de ces épingles à cheveux sont des composants d'éléments transposables.

Des études de Landgraf *et al.* et Piriyapongsa *et al.* décrivent les gènes de miARNs provenant d'ETs non autonomes [152, 103, 141] et des études récentes affirment que certains pré-miARNs partagent leurs séquences ou une partie importante de leurs séquences avec des ETs [150, 151, 171, 172]. Ces cas de pré-miARNs ont été annotés dans miRBase [98] et sont appelés des miARNs ET-dérivés [150].

2.3.3 Les piARNs, moyen de défense contre les éléments transposables

Les éléments transposables (ETs), outre qu'ils sont un moteur essentiel de l'évolution et de la diversité entre les espèces, peuvent être responsables de maladies si aucun contrôle n'est exercé dessus. Les piARNs sont l'un de ces moyens de contrôle.

Une des particularités des piARNs est qu'ils ressemblent étrangement aux séquences qu'ils régulent, à savoir les éléments transposables. Il est supposé que les piARNs sont d'anciennes séquences d'éléments transposables d'où leur regroupement sur le génome. En outre, les clusters de piARNs sont enrichis en séquences répétées, ce qui prouve que les piARNs sont des restes de transposons.

Récemment, des résultats ont rapporté que les protéines PIWI pouvaient également servir de nucléase pour couper les précurseurs de piARNs en 5' et donc produire de nouveaux piARNs matures, selon un cycle d'amplification appelé "amplification Ping-Pong" [113] (voir Figure 2.9).

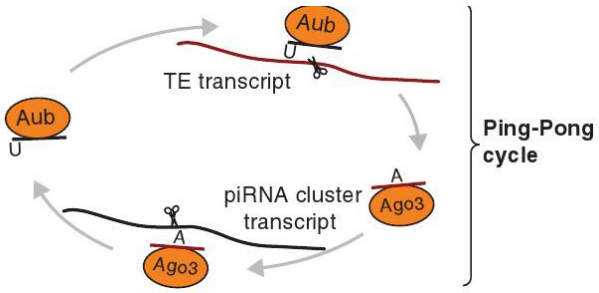

FIGURE 2.9 – Cycle d'amplification Ping-Pong des piARNs [113].

Le cycle d'amplification Ping-Pong permet, à partir d'un piARN, de produire un nombre important de nouveaux piARNs, selon le processus suivant : Après la formation du complexe Aubergine/piARN (tel que décrit en Section 2.2.4), celui-ci se fixe à un ET complémentaire au piARN, puis est coupé par l'Aubergine du ET à 10 nucléotides en 5' du piARN chargé. L'Argaunote 3 se lie ensuite au bout du ET coupé (cette séquence a fréquemment un uridine (U) en 1 et une adénine (A) en position 10). Ce nouveau complexe va alors agir sur un nouveau précurseur de piARN sortant du noyau et le couper au niveau du A en position 10, ce qui créé un U en position 1 du 5'. Ce brin va évoluer comme décrit en Section 2.2.4, formant ainsi le même complexe Aubergine/piARN observé au début du cycle.

Grâce à ce cycle, une cellule peut rapidement produire une grande quantité de piARNs ciblant un transposon bien particulier. Cette réponse rapide peut alors compenser la production rapide d'éléments transposables causée par exemple par un stress passager de l'organisme.

Chapitre 3

Prédiction de structures secondaires d'ARN

3.1 Introduction

La prédiction de structures secondaires d'ARNs est l'une des problématiques les plus importantes et les plus étudiées en bioinformatique des ARNs. La connaissance de la structure secondaire présente de nombreux intérêts biologiques. Tout d'abord elle constitue une étape simple et bien définie vers la résolution de la structure tertiaire plus complexe. Ensuite, la comparaison de structures secondaires d'ARNs permet de faire de la classification de séquences et de la phylogénie, car celles-ci sont souvent mieux conservées que les séquences elles-mêmes. Enfin, certains motifs de structure secondaire (tels que les motifs en épingle à cheveux ou hairpin) peuvent jouer des rôles importants dans certains processus de régulation.

3.1.1 Approches principales existantes

Il existe deux principales approches pour la prédiction de structure secondaire d'ARN : l'approche thermodynamique et l'approche comparative. Les premières méthodes *in silico* développées étaient basées sur l'approche thermodynamique. La seconde approche a été initialement utilisée pour prédire manuellement la structure secondaire de certains ARNs, avant d'être automatisée par différentes méthodes. Ainsi, les algorithmes de prédiction de structure secondaire d'ARNs existants sont basés sur l'une ou l'autre des deux approches, et certains d'entre eux combinent les deux approches.

Approche thermodynamique L'approche thermodynamique est basée sur le calcul de l'énergie libre des structures en utilisant des paramètres thermodynamiques définis expérimentalement [51, 127]. L'idée est que la structure réelle est celle d'énergie libre minimale. Les limites de cette approche sont dues en partie à l'incertitude du modèle énergétique utilisé. En effet, bien qu'il soit admis que la structure réelle possède une énergie relativement faible, elle n'est en général pas celle d'énergie la plus faible car le repliement de l'ARN est la plupart du temps aidé et guidé par un ensemble d'autres macromolécules. Une solution partielle à ce problème a été de développer des programmes permettant de calculer, non pas une seule structure, mais plusieurs, considérées comme les plus plausibles.

Le premier algorithme efficace basé sur cette approche a été proposé par Nussinov et Jacobson dans [142]. Il est basé sur la programmation dynamique, où tous les appariements possibles sont considérés, pour déduire ceux minimisant l'énergie thermodynamique globale. Une amélioration de cet algorithme a été proposée par Zuker, qui a développé le programme Mfold [126], de complexité en temps de $\mathcal{O}(n^3)$, celui-ci ayant ensuite été mis à jour pour permettre de retourner plusieurs structures sous-optimales [217], et une autre par Hofacker et al., qui ont développé l'algorithme RNAFold [78]. Mfold et RNAFold sont les deux logiciels les plus utilisés pour la prédiction de la structure secondaire d'une séquence donnée.

D'autres implémentations de l'approche thermodynamique autrement que par la programmation dynamique ont été proposées. On peut citer par exemple l'algorithme de Ninio [43], puis celui de Martinez [123, 123], qui recherche les hélices récursivement, en gardant à chaque étape les hélices formant une structure avec une énergie ne dépassant pas un certain seuil. Shapiro a ensuite développé une autre implémentation de l'approche thermodynamique, basée sur les algorithmes génétiques [169].

Approche comparative L'approche comparative est utilisée lorsque plusieurs séquences alignées homologues d'ARN, c'est-à-dire des séquences d'un même ARN appartenant à des espèces différentes, sont disponibles [85]. Elle consiste à rechercher des covariations entre les nucléotides de différentes séquences, permettant de maintenir des appariements et donc une structuration des séquences. Grâce à cette approche, les biologistes ont pu déterminer manuellement les structures secondaires de certains ARN ribosomaux, tels que l'ARNr 16S et l'ARNr 23S, dont les longueurs sont de l'ordre de quelques milliers de nucléotides [106, 140, 139, 67]. Des méthodes automatiques ont plus tard été proposées. Le premier algorithme, de complexité en temps d'exécution et en espace mémoire raisonnables, a été proposé dans [69], où les cinq plus plausibles structures secondaires communes à m séquences homologues de longueurs égales à n sont produites en temps égal à $\mathcal{O}(m \times n^2 + n^3)$, avec un espace mémoire égal à $\mathcal{O}(n^2)$. Une autre automatisation a été ensuite proposée dans [61], avec une approche similaire à celle de [69]. L'algorithme est basé sur la programmation dynamique, utilisant les SCFGs (Stochastic Context-Free Grammars). On peut également citer Pfold [96], basé sur les grammaires context-free, et ayant une complexité de $\mathcal{O}(n3)$.

Certaines méthodes combinent les deux approches thermodynamique et comparative. On peut citer RNAalifold [218], qui intègre l'information thermodynamique et phylogénétique pour prédire une structure secondaire commune d'un ensemble de séquences homologues avec une complexité en temps de $\mathcal{O}(n^3)$. D'autres méthodes plus récentes combinent les informations de covariations et thermodynamiques en utilisant les SVM (support vector machine) avec des vecteurs de caractéristiques combinant les informations de covariations et les informations thermodynamiques [209, 216].

Un problème important dans l'approche comparative est le fait que les résultats de prédiction dépendent fortement des séquences homologues utilisées et de la qualité de l'alignement. Afin d'éviter cette dépendance, certains algorithmes se proposent d'effectuer l'alignement des séquences en même temps que la recherche d'une structure secondaire commune. En raison de la complexité en temps, ils utilisent très peu de séquences. On peut citer CaRNAc [148], Foldalign [59], Dynalign [125, 72], PARTS [71] et RAF [42].

3.1.2 Problème de recherche de pseudonœuds

La plupart des algorithmes développés pour la prédiction de structures secondaires d'ARN ne permettent pas la recherche de pseudonœuds. Les principales raisons sont les complexités en temps trop élevées. Dans [119], il a été prouvé que le problème général de la prédiction de structures secondaires d'ARN contenant des pseudonœuds est NP-difficile pour une large classe de modèles raisonnables de pseudonœuds. Les rares algorithmes qui permettaient de prédire les pseudonœuds étaient limités à une catégorie restreinte de pseudonœuds. Par exemple, dans [21], une modélisation pour la recherche de pseudonœuds basée sur les Stochastic Context Free Grammars (SCFG) est proposée. Elle est limitée à la recherche de pseudonœuds dans des structures avec deux hélices et trois boucles simples, sur de très petites séquences, avec une complexité en $\mathcal{O}(n^3)$. Dans [170], un algorithmes génétique a été développé pour prédire la structure secondaire incluant certains types de pseudonœuds. Dans [159], un algorithme de programmation dynamique est présenté, prédisant certains types simples de pseudonœuds. Sa complexité est de $\mathcal{O}(n^6)$ en temps et $\mathcal{O}(n^4)$ en espace. Dans [119], un algorithme en $\mathcal{O}(n^5)$ en temps et $\mathcal{O}(n^3)$ en espace a été proposé, avec un modèle qui permet certains types de pseudonœuds. Dans [73], un algorithme de repliement avec une énergie libre minimale a été mis en place. Il nécessite $\mathcal{O}(mn^3)$ en temps et $\mathcal{O}(mn^2)$ en espace, où m est une constante dépendant de la liberté structurelle associée au pseudonœud. Il cherche seulement le type le plus simple de pseudonœuds, à savoir le H-type.

Certaines hypothèses stipulent que, pour des raisons cinétiques, la structure secondaire réelle a souvent une énergie libre minimale locale plutôt que globale [1]. Certains algorithmes prennent ainsi en compte ces caractéristiques cinétiques afin de minimiser l'énergie libre dans une zone locale. Plusieurs d'entre eux tentent de simuler les processus de repliement d'ARN de manière itérative par l'ajout de tiges au lieu de paires de bases [27, 213, 164]. La stratégie de recherche itérative des hélices permet ainsi de réduire l'espace de recherche et de traiter les structures avec des pseudonœuds.

3.1.3 Notre contribution

Nous avons développé, en collaboration avec Mireille Régnier de l'INRIA, un algorithme pour l'automatisation de l'approche comparative, approche qui était assez peu exploitée du point de vue informatique, mais qui semblait être prometteuse en terme d'efficacité des résultats. L'algorithme, appelé *DCfold* (Data Conquer Folding) est principalement basé sur une approche "diviser pour régner", où les hélices sont recherchées récursivement des plus "pertinentes" aux moins "pertinentes", cette pertinence étant définie par différents critères, dont la longueur et la covariation. Les particularités de cet algorithme est multiple :

- Il recherche la structure secondaire d'une séquence d'ARN donnée, appelée séquence cible, en tenant compte d'un ensemble de séquences homologues à cette séquence, appelées séquences tests. Dans la plupart des algorithmes existants basés sur l'approche comparative, on recherche plutôt une structure commune à toutes les séquences considérées.
- Il a besoin d'un nombre non élevé de séquences tests (autour de 4 séquences tests pour une séquence cible de quelques centaines de nucléotides).
- Il recherche sur la séquence cible les successions d'appariements qui forment les hélices de la structure, plutôt que de rechercher tous les appariements possibles, comme cela est fait dans la plupart des algorithmes de prédiction existants.
- La complexité en temps de *DCfold* est de $\mathcal{O}(n^2)$, alors qu'elle est de $\mathcal{O}(n^3)$ dans les autres algorithmes.

Cet algorithme a permis d'obtenir de bons résultats de prédiction sur différents ARNncs et a été publié dans la revue internationale Computers & Chemistry [179].

J'ai ensuite poursuivi ce travail avec Stéfan Engelen, dans le cadre de sa thèse effectuée sous mon encadrement de 2002 à 2006. Nous avons dans un premier temps développé un algorithme appelé *P-DCfold* (Pseudoknots Divide and Conquer Folding). Il s'agit d'un algorithme permettant de prédire la structure secondaire d'ARN incluant les pseudonœuds. Lorsque *P-DCfold* a été développé, il existait dans la littérature un nombre limité d'algorithmes intégrant la recherche de pseudonoeuds, et ces derniers avaient des complexités très élevées ($O(n^5)$ et plus) et ne recherchaient qu'un certain type de pseudonœuds. *P-DCfold* permet de rechercher tous les pseudonœuds, quels que soient leurs types, avec une complexité en temps de $\mathcal{O}(n^2)$, et donc en des temps très rapides, et avec de bons résultats de prédiction. La particularité de *P-DCfold* est qu'il recherche les pseudonœuds après avoir trouvé la structure de l'ARN sans pseudonœuds. En d'autres termes, il recherche chacune des deux (ou plus) hélices composant le pseudonœud à des étapes différentes. Notre algorithme a été présenté et publié dans la conférence internationale de Bioinformatique BIBE [177] à l'issue de laquelle l'article a été invité pour une publication dans la revue internationale IJAIT [178].

L'un des grands inconvénients de l'approche comparative est la dépendance des résultats par rapport aux séquences considérées et à la qualité de leur alignement. Certains algorithmes de la littérature proposent d'effectuer la prédiction de la structure commune en même temps que l'alignement. Cette approche est très intéressante mais a une complexité élevée, et les algorithmes existants ne considèrent que deux séquences. Nous avons pour notre part choisi de développer une nouvelle approche, qui consiste à sélectionner en amont de la prédiction les séquences homologues les plus adaptées pour la recherche de la structure conservée. Nous avons développé l'algorithme *SSCA* (Sequences Selection in Comparative Approach), qui utilise

des idées basées sur les modèles d'évolution des séquences d'ARN avec des contraintes de structure. Ce travail a été publié dans une conférence internationale [45], puis dans la revue internationale BMC Bioinformatics [46] dans sa version plus aboutie.

SSCA a été ensuite intégré à *P-DCfold* afin d'améliorer les résultats de prédiction et surtout afin d'avoir un algorithme qui ne soit pas sensible aux séquences homologues données en entrée. L'algorithme obtenu, appelé Tfold, prend en entrée un ensemble de séquences alignées, sélectionne les séquences les mieux alignées et les plus informatives pour la prédiction de la structure secondaire de la séquence cible choisie, puis lance la prédiction plusieurs fois avec des sous-ensembles de séquences tests différents. Les hélices trouvées par plus de 50% des prédictions sont sélectionnées, pour former la structure finale.

En marge de l'intégration de *SSCA*, plusieurs améliorations ont été apportées à Tfold par rapport à *P-DCfold* :
– Des critères thermodynamiques ont été intégrés pour la sélection des hélices.
– Les hélices sélectionnées peuvent présenter quelques renflements ou boucles internes.
– Plusieurs solutions de structures secondaires possibles peuvent être proposées pour un ARN donné.
– Une hélice ou plusieurs hélices connues peuvent être prises en compte dans l'algorithme.
Tfold a été comparé à différents logiciels existants ; il présente pour chaque ARN testé des sélectivités et sensibilités qui sont toujours dans les 3 meilleurs résultats, le plus souvent dans les 2 meilleurs. En particulier, pour les ARNs présentant des pseudonœuds, il est toujours meilleur. La force de Tfold est qu'il donne des résultats homogènes, toujours supérieurs à 80%, quelque soit l'ARN testé, tandis que les autres algorithmes peuvent donner de très bons résultats pour certains ARNs mais de très mauvais pour d'autres. De plus, plusieurs d'entre eux ne peuvent traiter que des séquences de taille limitée, ce qui n'est pas le cas de Tfold. Enfin, Tfold est très rapide, quelques secondes seulement pour une séquence d'ARN de taille moyenne (de plusieurs centaines de nucléotides). Ce travail a été publié dans la revue Nucleic Acids Res. [47].

3.2 Définitions

3.2.1 Définitions et représentations

Definition (Structure secondaire d'ARN) : *Une structure secondaire d'ARN est composée d'un ensemble d'hélices, de renflements et de boucles internes, multiples et terminales.*

Une hélice est définie par une *répétition palindromique* qui représente un type particulier de répétition dans la séquence, à savoir une répétition inverse et complémentaire (voir Figure 3.1).

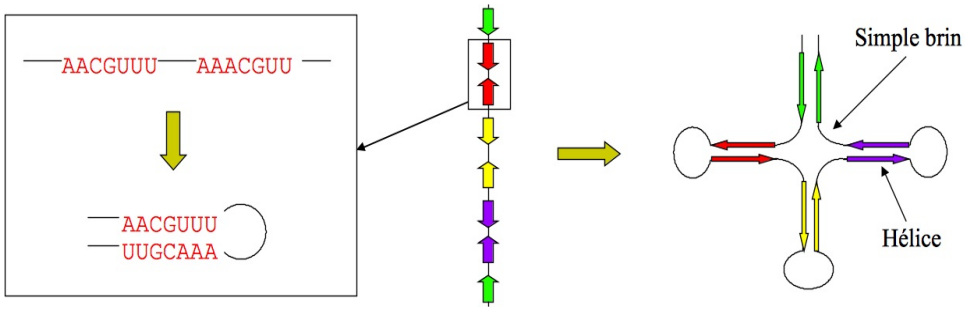

FIGURE 3.1 – Une structure secondaire résulte de l'appariement de répétitions inverses et complémentaires.

Definition (Hélice) :

Une hélice de longueur l est un couple de mots (p, p') tel que :
i) $|p| = |p'| = l$
ii) $p[k]R_c p'[l-k+1], \qquad \forall k, 1 \leq k \leq l,$
où R_c est la relation de complémentarité entre les nucléotides : AR_cU, GR_cC et GR_cU.

Une hélice de longueur l est donc une succession de l appariements.

Les hélices apparaissant dans une séquence peuvent être représentées par des arcs, chaque arc reliant les deux parties du hélice qu'il représente. Cette représentation en arcs est très utilisée pour schématiser la structure secondaire. La Figure 3.2 en montre un exemple.

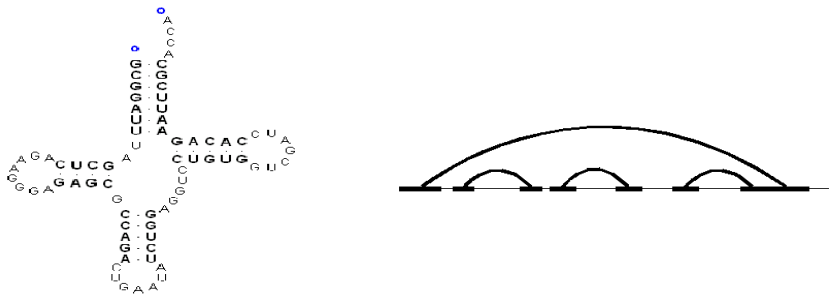

FIGURE 3.2 – Représentation par des arcs d'une structure secondaire d'ARN.

Definition (Hélice conservée) : *Etant donné un ensemble de séquences alignées, une hélice apparaissant dans chaque séquence à la même position alignée (éventuellement avec un certain décalage) est dite conservée.*

Une hélice (p, p') n'est pas nécessairement conservée avec les mêmes paires de bases. Certaines mutations peuvent se produire. Elles sont dites *compensées* lorsque l'appariement reste encore possible.

Definition (Mutation compensatoire) : *Etant donnée une séquence d'ARN, et étant donnée une paire de bases complémentaires (b_1, b'_1) dans cette séquence, celle-ci subit une mutation compensatoire en la paire (b_2, b'_2) si les bases b_2 et b'_2 sont également complémentaires.*

En d'autres termes, une mutation compensatoire est constituée de deux substitutions de nucléotides qui se compensent pour maintenir la complémentarité des bases. C'est donc une mutation double.

Definition (Mutation simple compensée) : *Nous dirons qu'un couple de bases (b_1, b'_1) a subi une mutation simple compensée lorsqu'une seule des deux bases b_1 ou b'_1 a muté, tout en maintenant la complémentarité.*

La mutation simple est liée à la double complémentarité des bases G et U. Elle apparaît donc dans le cas des couples (G, U), (G, C) ou (A, U).

Le phénomène de mutations compensatoires est important dans les ARNs, du fait qu'il permet de garder une structure secondaire similaire pour un ensemble de séquences d'espèces différentes, malgré les différences de nucléotides dues à des mutations que présentent ces séquences.

Definition (Hélices compatibles) :

Deux hélices (p, p') et (q, q') sont compatibles si elles se présentent disjointes ou imbriquées. Sinon si elles sont chevauchantes (ou entrelacées) elles sont dites incompatibles (voir Figure 3.3).

FIGURE 3.3 – Les trois cas de figure de deux hélices apparaissant dans la même séquence : disjointes (à gauche), imbriquées (au centre) et entrelacées (à droite).

Definition (pseudonœud) : *Un pseudonœud est une structure qui résulte de l'appariement de deux hélices entrelacées.*

Il existe certains pseudonœuds composés de plus de deux hélices entrelacées, comme c'est le cas de l'opéron α ARNm de l'*Escherichia coli*, présentant un pseudonœud composé de trois hélices entrelacées [57, 181, 182].

Definition (P-pseudonœud) : *Un P-pseudonœud est un pseudonœud composé de P hélices entrelacées.*

Definition (Complexité d'une structure secondaire) : *Une structure secondaire de l'ARN a une complexité C, $C > 0$, si elle contient au moins un C-pseudonœud et pas de $(C + k)$-pseudonœud pour tout $k > 0$. Lorsque C est égal à 1 la structure secondaire ne contient pas de pseudonœuds.*

La représentation par des arcs permet de mettre en évidence facilement la complexité des structures secondaires. En effet, une structure secondaire pouvant être représentée par des arcs sans aucun croisement ne possède pas de pseudonœuds et est de complexité 1 (voir Figure 3.4, gauche). Dans le cas contraire, la structure possède des P-pseudonœuds.. Un moyen pour déterminer la complexité d'une structure est de représenter les arcs engendrant des croisements dans un autre plan, afin qu'il n'y ait plus de croisement (Figure 3.4, centre et droite). Le nombre de plans nécessaires pour retirer tous les croisements représente la complexité de la structure secondaire modélisée. Cette notion appelée "*book-thickness*" (épaisseur du livre) est très utilisée pour représenter la complexité des structures secondaires d'ARN [74, 73].

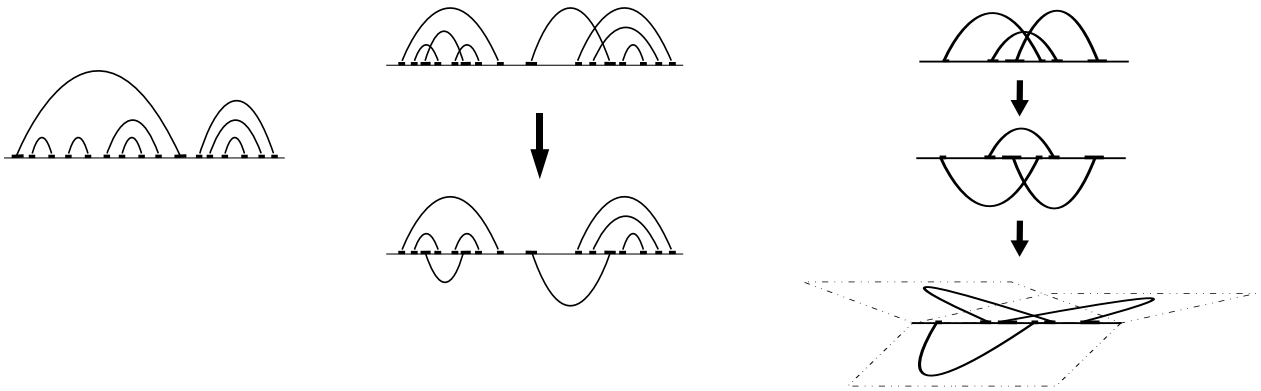

FIGURE 3.4 – Exemples de structures. Gauche : Structure de complexité 1 dont les arcs peuvent être représentés dans un seul plan sans croisement. Centre : Structure de complexité 2 dont les arcs se croisent, que l'on peut représenter sans croisement en utilisant deux plans. Droite : Structure de complexité 3 nécessitant trois plans pour une représentation sans croisement d'arcs.

3.2.2 Mesures utilisées pour l'évaluation des résultats de prédiction

Pour évaluer les méthodes permettant de prédire les structures secondaires d'ARN, les mesures les plus utilisées sont la sensibilité et la sélectivité. La sensibilité mesure la capacité à trouver les appariements d'une structure de référence. Une sensibilité de $0,90$ signifie que 90% des appariements de la structure de référence sont trouvés. La sélectivité, appelé également PPV ("Prédictive Positive Value"), représente la probabilité qu'un appariement prédit appartienne à la structure de référence. Une sélectivité de $0,90$ signifie que 90% des appariements prédits sont de vrais positifs et 10% sont des faux positifs. Les mesures de sensibilité et de sélectivité sont données par les équations suivantes :

$$Sensibilite = \frac{TP}{TP + FN} \qquad Selectivite = \frac{TP}{TP + FP}$$

où TP est le nombre d'appariements correctement prédits (vrais positifs), FN est le nombre d'appariements non prédits (faux négatifs) et FP est le nombre d'appariements prédits qui ne figurent pas dans la structure (faux positifs).

Un troisième critère permet d'évaluer simultanément la sensibilité et la sélectivité. Ce critère, appelé MCC, pour "coefficient de corrélation de Mathews" [8], est calculé comme suit :

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

où TN, le nombre de vrais couples négatifs, est égale à : $(n*(n-1)/2) - TP - FN - FP$ avec $n*(n-1)/2$, représentant tous les appariements possibles dans une séquence de taille n. Ainsi, TN représente tous les appariements possibles, moins les vrais positifs, les faux négatifs et les faux positifs. MCC varie de -1 à 1, 1 correspondant à des prévisions qui correspondent complètement à la structure de référence.

Parce les appariements faux positifs ne sont pas nécessairement faux, Gardner et Giegerich ont introduit dans [54] une valeur e représentant le nombre de paires de faux positifs qui ne sont pas en conflit avec les appariements de la structure de référence. La sélectivité et le MCC deviennent alors :

$$Selectivite = \frac{TP}{TP + (FP - e)} \qquad MCC = \frac{(TP * TN) - (FP - e) * FN}{\sqrt{(TP + (FP - e))(TP + FN)(TN + (FP - E))(TN + FN)}}$$

3.3 *DCfold* : Prédiction de structures secondaires d'ARN basée sur l'approche comparative

Nous avons développé un algorithme pour l'automatisation de l'approche comparative, approche qui était encore assez peu exploitée du point de vue informatique. L'algorithme, appelé DCfold (Data Conquer Folding) est principalement basé sur une approche "diviser pour régner", où les hélices sont recherchées récursivement des plus "pertinentes" aux moins "pertinentes", cette pertinence étant définie par différents critères, dont la longueur et la covariation.

3.3.1 Notre approche

Nous considérons un ensemble E de séquences alignées d'un même ARN d'espèces appartenant à une même famille (séquences homologues), et nous distinguons parmi ces séquences une séquence cible pour laquelle on veut prédire la structure secondaire.

Notre approche est basée essentiellement sur une recherche de motifs, et consiste à rechercher les hélices de la structure secondaire Les hélices étant définies par des répétitions inverses et complémentaires, la première étape de notre algorithme consiste à rechercher ces répétitions dans la séquence cible. Puis une étape de comparaison avec les autres séquences est réalisée pour déduire les hélices conservées.

Les hélices sont recherchées récursivement, des plus "pertinentes" au moins "pertinentes", en utilisant l'approche "diviser pour régner". Une hélice sélectionnée est considérée comme un "point d'ancrage" permettant de subdiviser la séquence initiale en deux sous-séquences indépendantes, celle qui lui est interne et celle résultant de la jonction des deux sous-séquences qui lui sont externes. L'ordre dans lequel sont sélectionnées les différentes hélices dépend de deux critères, le critère de longueur et le critère de nombre de mutations compensatoires (voir Section 3.2.1). Nous considérons en effet que les hélices sont d'autant plus "pertinentes" que leur taille est grande et que le nombre de mutations associées est élevé.

En résumé, le principe algorithmique de notre recherche des hélices est le suivant :
- Rechercher dans la séquence cible S les hélices vérifiant certains critères de sélection prédéfinis.
- Comparer les hélices trouvés avec les séquences tests, pour sélectionner celles qui apparaissent dans toutes les séquences et qui vérifient des critères supplémentaires.
- Ré-itérer le processus ci-dessus sur des sous-séquences de S suffisamment longues (pour contenir des hélices) déduites de la subdivision de S à partir des hélices trouvés initialement, en considérant de nouveaux critères de sélection.

3.3.2 Critères de sélection des hélices

Critère de longueur Le critère initial de sélection des hélices est leur longueur. En effet, dans une séquence donnée, si les petits mots apparaissent "presque" toujours, l'apparition d'un long mot n'est pas, a priori, due au hasard, d'où la (possible) "pertinence" de celui-ci. Dans [50], il a été montré que dans une séquence de longueur n, presque tous les mots de longueur inférieure à $log_k n$ où k est la taille de l'alphabet, apparaissent. Nous évaluons ainsi la longueur minimale des hélices "pertinentes" apparaissant dans une séquence de nucléotides de longueur n à $log_4 n$. Pour une séquence d'un millier de nucléotides, ceci se situe autour d'une longueur de 6.

Etant donné un ensemble de séquences, nous recherchons initialement dans la séquence cible choisie les hélices de longueur supérieure ou égale à $log_4 n$, n étant la longueur de cette dernière. Les hélices trouvées sont ensuite comparées avec les séquences tests. Nous sélectionnons celles qui sont conservées avec cette longueur minimale de $log_4 n$.

Néanmoins, comme nous pouvons le voir sur l'exemple donné dans la Table 3.1, le critère de longueur des

hélices est un critère important, mais non suffisant.

taille min hélice	nbre hélices trouvées	nbre vrais positives	% vrais positives
6	48	18	37%
7	22	12	54%
8	5	4	80%
9	2	2	100%
10	1	1	100%

TABLE 3.1 – Variation du pourcentage des hélices par rapport à l'ensemble des hélices vérifiant une longueur donnée, dans la structure secondaire de l'ARN 16S de E.coli.

Critère de covariation Nous avons défini un critère supplémentaire de sélection des hélices, que nous combinons avec le critère de longueur. Il s'agit du critère de nombre de mutations. Il s'agit plus exactement des mutations compensatoires (composées d'une double mutation) ou des mutations simples compensées, qui permettent de maintenir les appariements (voir Section 3.2.1). Une hélice est d'autant plus pertinente que le nombre de mutations compensatoires qu'elle a subies est plus élevé.

Le calcul du nombre de mutations est réalisé lors de la phase de comparaison. Etant donnée une hélice conservée, le nombre de mutations N_{mut} est calculé comme suit : $N_{mut} = 2 * N_{comp} + N_{cons} - N_{err}$, où N_{comp} est le nombre de mutations compensatoires, N_{cons} le nombre de mutations simples compensées et N_{err} le nombre d'erreurs (appariements non conservés). Avec cette équation, nous favorisons les hélices conservées avec un haut niveau de covariation (plusieurs mutations compensatoires). Nous prenons en compte les hélices conservées avec des erreurs mais celles-ci sont discriminées.

Critère de sélection global Nous avons défini un critère de sélection global, appelé $LongMut$, qui combine les critères de longueur et de nombre de mutations ($LongMut = l + N_{mut}$, avec l la longueur de l'hélice). Ce critère est calculé pour chaque hélice conservée.

La sélection des hélices se fait donc sur la base de la valeur de leur paramètre $LongMut$, en favorisant celles pour lesquelles cette valeur est grande. Ainsi, une hélice est sélectionnée si elle vérifie l'équation suivante:

$$LongMut \geq 2 * lmin \tag{3.1}$$

avec $lmin$ la longueur minimale que doit vérifier l'hélice, à savoir $log_4 n$, n étant la longueur de la séquence. Ainsi, afin de vérifier l'équation 3.1, les hélices de longueur $lmin$ doivent posséder au moins $lmin$ mutations compensatoires. Si elles en possèdent moins, elles doivent être plus longues pour vérifier cette équation.

Parmi les hélices éliminées dans l'une ou l'autre des deux phases de sélection, certaines peuvent être des vraies positives. Elles sont retrouvées lors de la ré-itération du processus de recherche sur les sous-séquences. En effet, à chaque itération, les sous-séquences traitées sont de tailles de plus en plus petites, entraînant la réduction du seuil de sélection des points d'ancrage.

3.3.3 Recherche des hélices conservées

Etape de recherche dans la séquence cible La première étape de recherche des hélices consiste à rechercher les hélices qui apparaissent dans la séquence cible S de longueur n et qui vérifient le critère de longueur minimale $lmin$. Pour cela, une matrice comparant la séquence S avec son inverse est utilisée. Elle

est construire de la manière suivante :

$$M[0,i] = M[j,0] = 0 \quad et \quad M[i,j] = \begin{cases} M[i-1,j-1]+1 & \text{si S(i) et S(j) forment un appariement} \\ 0 & \text{sinon} \end{cases}$$

pour tout $i \leq n$ et $j \leq n$, puis sont sélectionnées les répétitions reconnues par les cases contenant des valeurs supérieures ou égales à $lmin$.

Etape de comparaison avec les séquences tests L'étape de comparaison consiste ensuite à vérifier, pour chaque répétition palindromique (p, p') de longueur l trouvée à une position (deb, fin) dans S (deb et fin sont les positions respectives de p et p' dans S), s'il apparaît, dans chacune des séquences tests, une répétition palindromique (q, q') de longueur l', $l' \geq lmin$, à la position $(deb \pm k, fin \pm k)$, k étant un éventuel décalage ($0 \leq k \leq l - lmin$). Notons que (q, q') peut être différent de (p, p'), une hélice conservée n'étant pas obligatoirement définie par la même composition de bases. Les hélices conservées dans toutes les séquences sont sélectionnées si elles vérifient le critère de sélection (Equation 3.1).

Certaines zones peuvent être fortement variables, c'est à dire présentant un nombre important de mutations non compensées. Ces mutations sont souvent de type insertion-suppression et la variabilité des séquences est donc reflétée par la différence de leurs longueurs dans la zone considérée. La solution choisie pour traiter ce problème consiste à subdiviser l'ensemble initial des séquences en plusieurs sous-ensembles, chacun regroupant les séquences qui sont proches, c'est-à-dire celles qui possèdent des longueurs voisines dans la zone considérée.

3.3.4 Approche "diviser pour régner"

Notre algorithme est basé sur une approche "diviser pour régner", où les hélices sont recherchés récursivement. En effet, lorsque les pseudonœuds ne sont pas considérés, une hélice trouvée permet de subdiviser une séquence en deux sous-séquences, où d'autres hélices peuvent être recherchées (voir Figure 3.5).

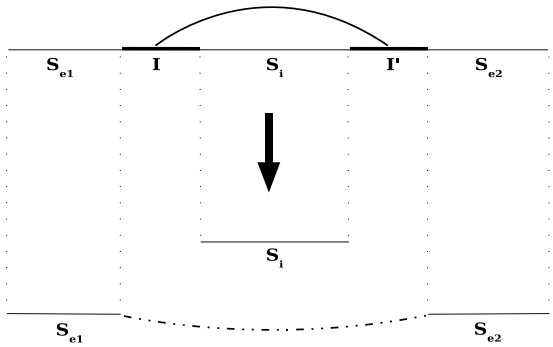

FIGURE 3.5 – L'approche "diviser pour régner" appliquée à une hélice (I, I'). L'hélice (I, I') permet de subdiviser la séquence en deux sous-séquences sur lesquelles d'autres hélices peuvent être recherchées : la séquence interne S_1 et la concaténation des deux sous-séquences externes $S_{e1}S_{e2}$.

Ensemble valide de points d'ancrage - Notion de compatibilité Lorsqu'un ensemble d'hélices (hélices conservées vérifiant les critères de sélection) a été sélectionné, il est utilisé comme ensemble de points d'ancrage permettant de subdiviser la séquence en plusieurs sous-séquences sur lesquelles est ré-itérée la recherche d'autres hélices. Néanmoins, afin que la subdivision de la séquence soit possible, les hélices sélectionnées doivent constituer ce que nous appelons un *ensemble valide de points d'ancrage*.

Definition (Ensemble valide de points d'ancrage) : *Un ensemble d'hélices sélectionnées constitue un ensemble valide de points d'ancrage si elles sont toutes compatibles entre elles.*

Ainsi, la validation d'un ensemble de points d'ancrage se fait par la vérification de leur compatibilité mutuelle. Nous avons développé une procédure de traitement d'incompatibilité qui consiste à valider parmi les hélices incompatibles entre elles celles qui sont supposées être les plus pertinentes. Elle se base, pour cela, sur la valeur du paramètre $LongMut$ associé à chacune de ces hélices. Elle privilégie celles pour lesquelles ce paramètre est le plus élevé, en éliminant celles qui sont incompatibles avec ces dernières.

Subdivision de la séquence initiale pour la recherche de nouveaux points d'ancrage Etant donné un ensemble initial de points d'ancrage trouvés, il s'agit de rechercher de nouveaux points d'ancrage qui vérifient d'autres critères et qui sont compatibles avec les précédents. Pour cela, la séquence initiale est subdivisée en différentes sous-séquences, déduites de cet ensemble.

Afin d'éviter une redondance dans le traitement des différentes sous-séquences, la subdivision de la séquence se fait comme suit : étant donnée une liste de points d'ancrage, celle-ci est triée par ordre croissant sur la position de fin d'occurrence des points d'ancrage, puis les sous-séquences associées aux différents points d'ancrage sont traitées dans l'ordre de leur apparition dans la liste triée. Le but est de traiter en premier les sous-séquences associées aux hélices les plus internes, jusqu'à arriver à la séquence globale dépourvue des sous-séquences traitées. Un exemple est donné en Figure 3.6.

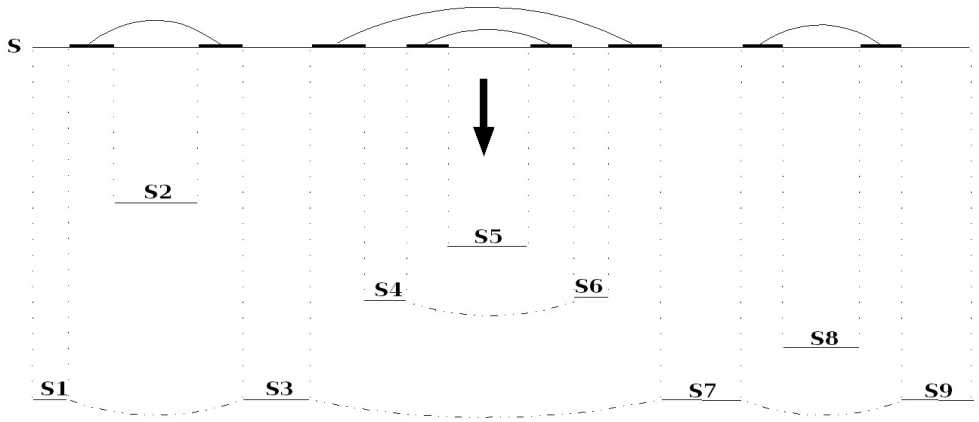

FIGURE 3.6 – L'approche "diviser pour régner" appliquée sur un ensemble d'hélices sélectionnées: les sous-séquences sont traitées dans l'ordre suivant : $S2$, $S5$, $S4S6$, $S8$ et enfin $S1S3S7S9$.

3.3.5 Algorithme

Etant donnée une séquence cible S et un ensemble de séquences homologues A, l'algorithme *DCfold* recherche les hélices apparaissant dans S et conservées dans toutes les séquences de A et sélectionne celles vérifiant le critère de longueur et de covariation, et formant une liste valide de points d'ancrage. A partir de cette liste la recherche d'autres points d'ancrage est lancée récursivement sur les sous-séquences déduites de ces derniers.

Nous donnons dans la Figure 3.7 la procédure *Recherche_hélices* qui recherche l'ensemble des hélices dans une séquence cible donnée S et un ensemble A de séquences alignées :

Ce processus récursif peut s'arrêter pour plusieurs raisons :

Procédure *Recherche_hélices* (S : séquence cible, A : Alignement de séquences homologues, Lg : Liste globale des hélices prédites)
Début
Soit n la taille de la séquence S
Soit $lmin$ la longueur minimale des hélices recherchées : $lmin = log_4(n)$
$L = \emptyset$
si $n > 10$ **alors**
Chercher les hélices conservées dans A de longueur $long$ tel que $long \geq lmin$
si aucune hélice n'est trouvée
alors si zone à forte variabilité **alors** Traiter la variabilité
fsi
fsi
Calculer pour chaque hélice trouvée le nombre de mutations N_{mut}
Sélectionner les hélices vérifiant $LongMut \geq 2 \times lmin$
S'il y a des hélices incompatibles entre elles, traiter l'incompatibilité
Mettre dans la liste L les hélices qui sont toutes compatibles entre elles
$Lg = Lg \cup L$
Pour chaque hélice (p, p') de la liste L triée
faire Soit la sous-séquence S_i dans S délimitée par (p, p')
Recherche_hélices (S_i, A, L_g)
fait
Fin

FIGURE 3.7 – Procédure de recherche des hélices structurantes

– La sous-séquence est trop petite pour contenir une hélice. La taille minimale d'une hélice est de trois appariements et celle d'une boucle terminale est de quatre nucléotides. Ainsi, la taille minimale d'une séquence pouvant contenir des hélices est de 10.
– Aucune hélice n'est trouvée. Ceci peut être dû à une *forte variabilité* de la zone considérée, c'est-à-dire à la présence d'un nombre important de mutations, autres que compensatoires, ou au contraire à une trop grande conservation.
– L'ensemble des points d'ancrage sélectionné est non valide et le traitement de l'incompatibilité échoue.

Complexité de l'algorithme *DCfold* a une complexité en $\mathcal{O}(kmn^2)$, où n est la longueur de la séquence cible, m ($m << n$) le nombre de séquences tests et k ($k << n$) la profondeur de la structure secondaire, c.a.d. le nombre d'étapes de récursivité nécessaires pour prédire la structure secondaire.

La procédure *Recherche_hélices* a une complexité en $\mathcal{O}(mn^2)$ en temps car la recherche des hélices dans la matrice nécessite de tester la moitié des cases de la matrice, soit $\frac{n^2}{2}$ cases (la matrice est symétrique) et, dans le pire des cas, si tous les nucléotides sont appariés, on doit lancer la phase de comparaison et analyser pour chaque nucléotide de la séquence cible, un appariement par séquence test, soit m appariements. De plus, à chaque étape de la récursivité de la recherche des hélices, nous recherchons des points d'ancrages dans des sous-séquences non chevauchantes. Ainsi, la longueur totale des ces sous-séquences ne peut pas dépasser celle de la séquence initiale et est forcement inférieure à n. S'il y a k étapes de récursivité, on répète k fois une recherche ayant pour complexité $\mathcal{O}(mn)$, ce qui nous donne une complexité de l'algorithme en $\mathcal{O}(kmn^2)$. Les nombres k et m étant très inférieurs à n (la valeur maximale de k est de 7 (dans le cas de l'ARN 23S) et la valeur utilisée pour m est autour de 4), on peut approximer la complexité à $\mathcal{O}(n^2)$.

3.3.6 Conclusion

DCfold se démarque des autres algorithmes de prédiction de structure secondaire d'ARN par sa faible complexité en temps, qui est de $\mathcal{O}(n^2)$, les algorithmes existants dans la littérature ayant des complexités souvent de $\mathcal{O}(n^3)$. Cette faible complexité est due à deux raisons principales :

– Le plupart des méthodes existantes recherchent tous les appariements possibles, alors que dans *DCfold*, ce sont les hélices, constituées d'une suite d'appariements d'une certaine longueur, qui sont recherchées.
– L'approche "diviser pour régner" utilisée pour la recherche des hélices permet de ne pas explorer tout l'espace des solutions.

DCfold présente par ailleurs d'autres particularités telles que l'utilisation d'un nombre non élevé de séquences tests (autour de 4 séquences tests pour une séquence cible de quelques centaines de nucléotides), et la recherche de la structure secondaire d'une séquence d'ARN donnée, appelée séquence cible, en tenant compte d'un ensemble de séquences homologues à cette séquence, appelées séquences tests. Dans les autres algorithmes existants basés sur l'approche comparative, on recherche une structure commune à toutes les séquences considérées, et certains d'entre eux déduisent ensuite la structure de l'une des séquences.

Cet algorithme, publié dans la revue internationale Computers and chemistry [179], avait permis d'obtenir de bons résultats de prédiction sur différents ARNs non-codants. Néanmoins, il présentait plusieurs limites dont la plus importante était l'exclusion des pseudonœuds dans la structure secondaire prédite.

3.4 *P-DCfold* : Prédiction de pseudonœuds

Dans le cadre de la thèse de Stéfan Engelen, nous avons développé un algorithme appelé *P-DCfold* (Pseudoknot Divide and Conquer Folding) pour la prédiction de structure secondaire d'ARN incluant les pseudonœuds, algorithme permettant d'identifier tous les types de pseudonœuds avec une complexité de $\mathcal{O}(n^2)$.

3.4.1 Notre approche

La particularité de *P-DCfold* est qu'il recherche les pseudonœuds après avoir trouvé la structure de l'ARN sans pseudonœuds, avec la même approche décrite plus haut (*DCfold*). En d'autres termes, il recherche chacune des deux (ou plus) hélices composant le pseudonœud à des étapes différentes.

En effet, si on considère une séquence dont la structure sans pseudonoeuds a été prédite, l'espace des possibilités pour ajouter à cette structure des pseudonoeuds a été réduit car on a imposé des contraintes à la structure. Il suffit alors de rechercher des hélices compatibles entre elles, mais incompatibles avec celles précédemment trouvées. Ces hélices forment alors des pseudonoeuds avec les hélices initialement trouvées.

L'algorithme de recherche des pseudonoeuds est donc une itération de la recherche des hélices compatibles. Il consiste d'abord à rechercher toutes les hélices compatibles. La séquence est alors privée des sous-séquences correspondant à ces hélices et la recherche des hélices compatibles est alors relancée sur cette nouvelle séquence. Ainsi, comme on utilise les mêmes critères de sélection lors des deux étapes de recherche, on ne peut trouver lors de la seconde étape que des hélices incompatibles avec celles trouvées lors de la première étape. On peut généraliser ce raisonnement en itérant le processus tant que l'on trouve de nouvelles hélices. Ainsi, si la séquence possède des P-pseudonœuds importants pour la structure générale de l'ARN, ils devraient vérifier nos critères de sélection (Equation 3.1). Le nombre d'itérations nous donne alors la complexité C de la structure de l'ARN considéré (voir Section 3.2.1). Cette itération de la recherche des hélices compatibles permet de trouver tous les types de pseudonoeuds avec une complexité algorithmique du même ordre que celle de l'algorithme *DCfold*.

3.4.2 Algorithme

L'extension de *P-DCfold* est comme suit : lorsque la procédure *Recherche_hélices* de *DCfold* s'arrête, elle est relancée sur la séquence initiale sans les sous-séquences correspondant aux hélices sélectionnées.

Etant donnée une séquence S donnée, *Recherche_hélices* trouve une liste $L1$ de toutes les hélices compatibles qui satisfont nos critères de sélection. Relancer *Recherche_hélices* sur S dépourvue des sous-séquences correspondant aux hélices de $L1$ (séquence S') permet de trouver une autre liste $L2$ de toutes les hélices compatibles entre elles et qui ne sont pas compatibles avec les hélices de $L1$. Par conséquent, une hélice de $L2$ va former un 2-pseudonœud avec une hélice de $L1$. Puis relancer à nouveau *Recherche_hélices* sur S' dépourvue des sous-séquences correspondant aux hélices de $L2$ permet de trouver une troisième liste $L3$ d'hélices compatibles entre elles et qui ne sont pas compatibles avec les hélices de la liste $L1$ et avec les hélices de la liste $L2$. Par conséquent, une hélice de $L3$ formera avec une hélice de $L1$ et une hélice de $L2$ un 3-pseudonœud. Et ainsi de suite, jusqu'à ce qu'aucune hélice ne soit trouvée. Donc si *Recherche_hélices* est lancée C fois, la structure secondaire est trouvée avec une complexité égale à C (voir Section 3.2.1). Par conséquent, le principe de l'algorithme est de rechercher les pseudonœuds en plusieurs étapes, chaque hélice du pseudonœud étant sélectionnée dans une étape différente.

La procédure *Recherche_Toutes_hélices*, qui recherche les hélices dont celles formant des pseudonœuds, et basée sur la procédure *Recherche_hélices* définie dans la Section 3.3.5, est donnée en Figure 3.8.

Procédure *Recherche_Toutes_hélices* (S : séquence cible, A : Alignement de séquences homologues)
Début
$L_{all} = \emptyset$ * L_{all}: liste globale des hélices structurants
$n = |S|$ * n : taille de la séquence cible S
$C = 0$ * C : la complexité de la structure secondaire
$L_g = \emptyset$
Recherche_hélices (S, A, L_g)
$L_{all} \leftarrow L_{all} \cup L_g$
Tans que ($L_g \neq \emptyset$)
  faire $C = C + 1$;
    Soit S_g la séquence globale S sans les sous-séquences associées aux hélices de L_{all}
    $L_g = \emptyset$
    Recherche_hélices (S_g, A, L_g)
    $L_{all} \leftarrow L_{all} \cup L_g$;
  fait
Retourner (C, L_{all})
Fin

FIGURE 3.8 – Procédure de recherche des hélices de la structure secondaire d'ARN, y compris celles formant des pseudonœuds.

Complexité de l'algorithme P-DCfold a une complexité en $\mathcal{O}(C \times n^2)$, où n est la longueur de la séquence cible et C la complexité de la structure prédite. En effet, le processus récursif de recherche des hélices structurants est réitéré autant de fois que le degré de complexité C de la structure. En d'autres termes, la procédure *Recherche_hélices*, de complexité $\mathcal{O}(n^2)$ (voir Section 3.3.5) est itérée C fois. Le nombre C étant très inférieur à n (la complexité C des exemples les plus courants d'ARNs est égale à 1 ou 2), on peut approximer la complexité à $\mathcal{O}(n^2)$.

3.4.3 Résultats

Pour illustrer l'efficacité de *P-DCfold* à prédire la structure secondaire d'ARNs comprenant des pseudonœuds mais aussi d'ARNs ne comprenant pas de pseudonœuds, nous donnons ici les résultats donnés par *P-DCfold* sur les ARNs suivants : ARNtm, RNAse P, SRPRNA, u1RNA et 5S RNA (ces ARNs sont décrits dans la Section 2.2). Les trois premiers contiennent des pseudonœuds : ARNtm en contient quatre, RNAse P en contient deux, et SRPRNA en contient un. Les structures de ces différents ARNs sont données en Figure 3.9. Les hélices faux positives (prédites par P-DCFold mais non existantes dans la structure de référence) sont mentionnées par des flèches et les hélices faux négatives (hélices de la structure qui n'ont pas été prédites par P-DCFold) sont mentionnées en gras et encadrées.

Résultats sur l'ARNtm : Nous avons extrait du site tmRDB [191] cinq séquences, à savoir *Escherichia coli*, *Shewanella putrefaciens Aquifex aeolicus*, *Thermotoga maritima* et *Enterococcus faecalis*. Nous avons recherché la structure secondaire de la séquence d'*Escherichia coli*. La structure secondaire prédite par *P-DCfold* correspond à la structure connue (voir Figure 3.9, en haut à gauche). En effet, *P-DCfold* ne trouve pas d'hélices faux positives et seulement trois hélices faux négatives. Les quatre pseudonœuds de la structure ont été détectés avec succès.

Résultats sur le RNase P : Nous avons appliqué notre algorithme sur la séquence du RNase P de *Escherichia coli* en utilisant les quatre séquences test suivantes : *Desulfovibrio desulfuricans*, *Rhodospirillum rubrum*, *Streptomyces bikiniensis* et *Deinococcus radiodurans*. Notre algorithme détecte presque toutes les hélices (voir Figure 3.9, en haut à droite), dont les deux pseudonœuds. Aucune hélice faux positive n'a été

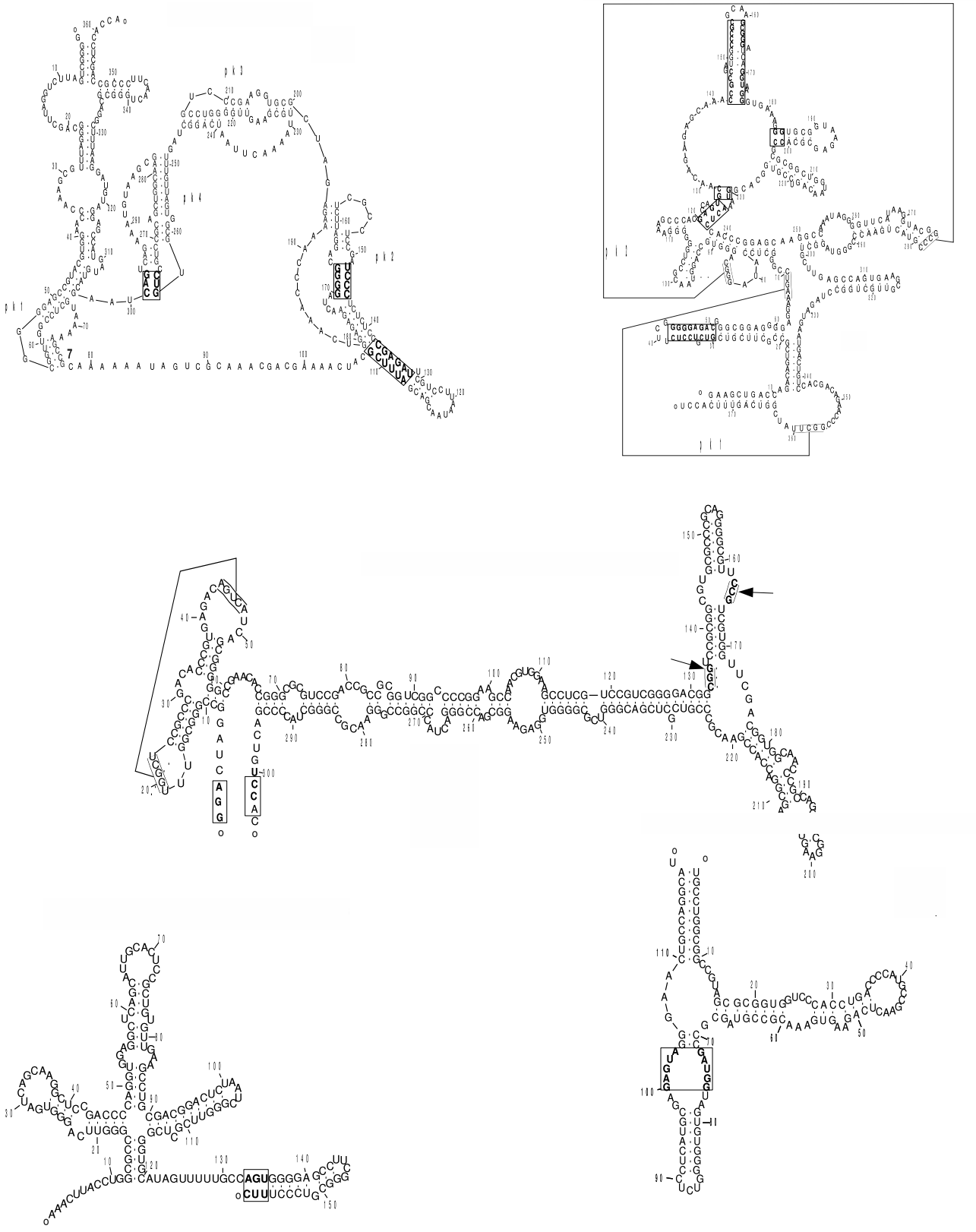

FIGURE 3.9 – Exemples de structures secondaires prédites par *P-DCfold* : ARNtm d'*Escherichia coli* (en haut à gauche), RNase P d'Escherichia coli (en haut à droite), ARN SRP de Halobacterium halobium (centre), ARN u1 d'Echinococcus multilocularis (en bas à gauche), et ARN 5S d'Escherichia coli (en bas à droite). Les parties encadrées correspondent à des hélices non prédites par *P-DCfold*. Les parties reliées correspondent aux pseudonœuds, tous prédits par *P-DCfold*. Enfin, les deux sous-séquences fléchées de l'ARN SRP correspondent à une hélice prédite par *DCfold* et donc à une hélice faux positive.

sélectionnée. Il y a cinq hélices faux négatives, mais parmi elles, trois sont des extensions d'hélices prédites.

Résultats sur le SRPRN : Nous avons appliqué notre algorithme sur SRPRNA de *Halobacterium halobium* en utilisant quatre séquences tests : *Haloferax volcanii, Methanococcus jannaschii, Methanothermus fervidus* et *Staphylococcus epidermidis*. La structure réelle de SRPRNA est composée de vingt hélices et notre prédiction n'a retourné qu'une seule faux positive et une faux négative (voir Figure 3.9, au centre). *P-DCfold* a permis de trouver le pseudonœud de la structure.

Résultats sur l'ARN u1: Nous avons prédit la structure du ARN u1 de *Echinococcus multilocularis*. Les séquences test utilisées pour cette prédiction et fournies par la base de données uRNA [219] sont de *Drosophila melanogaster, Caenorhabditis elegans, Physarum polycephalum* et *Tetrahymena thermophila*. La prédiction a permis de trouver les dix hélices de la structure (voir Figure 3.9, en bas à gauche). Il y a seulement une extension de la dernière hélice qui n'est pas trouvée par notre algorithme. En outre, *P-DCfold* n'a pas prédit de pseudonœuds, et donc de faux pseudonœuds (ARN u1 n'en contient aucun).

Résultats sur l'ARN 5S : Les séquences utilisées pour la prédiction sont fournies par la base de données 5S ribosomal RNA [176]. Nous avons prédit la structure de l'ARN 5S d'Escherichia coli avec les séquences test de *Helicobacter pylori, Clostridium carnis, Cytophaga aquatilis* et *Borrelia burgdorferi*. Une hélice de la structure est composée de couples non canoniques et n'a pas pu être trouvée par P-DCfold. Toutes les autres hélices sont bien trouvées par notre algorithme (voir Figure 3.9, en bas à droite).

3.4.4 Conclusion

L'algorithme *P-DCfold* permet de prédire la structure secondaire d'ARN incluant les pseudonœuds avec une complexité très avantageuse, de seulement de $\mathcal{O}(n^2)$, tout en assurant des résultats de prédiction très satisfaisants, alors que tous les autres algorithmes existants dans la littérature qui permettent la recherche de pseudonœuds, y compris les algorithmes plus récents (voir Section 3.6), ont une complexité dans le meilleur des cas en $\mathcal{O}(n^3)$. Comme nous avons pu le voir avec les tests effectués sur différents exemples d'ARNs, *P-DCfold* prédit efficacement les structures secondaires des ARNs avec ou sans pseudonœuds. Dans presque tous les cas, la complexité de la structure secondaire a été bien prédite : complexité 2 pour l'ARNtm, l'ARNase P et l'ARN SRP et complexité 1 pour l'ARN u1 et l'ARN 5S. De plus, les différentes structures sont prédites en des temps très rapides. Par exemple, sur la structure ARNtm, le temps d'exécution est moins de deux secondes, de même pour la structure RNase P. Ce travail a été publié dans la conférence internationale de bioinformatique BIBE [177] puis dans la revue internationale IJAIT [178].

Un point important est que la recherche de pseudonœuds n'influe pas sur la qualité globale des prédictions, ce qui n'est pas le cas de la plupart des algorithmes existants (voir Section 3.6). Les différentes structures sont globalement correctement prédites par *P-DCfold*. Une seule hélice faux positive a été prédite pour les cinq exemples d'ARNs que nous avons testés. En fait, de manière générale, notre algorithme prédit très peu voir quasiment pas d'hélices faux positives. La raison est que nous avons choisi de mettre en place des critères très sélectifs. De plus, il détecte presque toutes les hélices de la structure secondaire. Les rares exceptions concernent les régions à forte variabilité ou les régions hautement conservées, en d'autres termes sans mutations. Par exemple, dans la structure ARNtm, entre les trois hélices non détectées, deux correspondent à une région avec une forte variabilité (les hélices n'apparaissent pas dans l'une des séquences de test considérées) et une présente un mis-appariement dans l'une des séquences. Dans la structure RNase P, parmi les quatre hélices non détectées, deux correspondent à une région avec une forte variabilité et deux à une région sans mutations.

Ceci nous amène à aborder un problème important dans ce type d'algorithme, à savoir le problème de choix des séquences homologues. En effet, selon les séquences choisies, les résultats peuvent différer. En l'occurrence, des hélices peuvent ne pas être trouvées si les séquences sont trop proches (forte conservation) ou trop éloignées (forte variabilité) ou si les séquences sont mal alignées. Dans la section 3.5, nous montrons comment nous avons traité ce problème et les solutions que nous avons proposées.

3.5 *SSCA* : Sélection de séquences homologues pour l'approche comparative

De manière générale, la qualité des résultats obtenus avec l'approche comparative est meilleure par rapport à celle des résultas obtenus par l'approche thermodynamique [54]. Avec l'augmentation considérable du nombre de séquences disponibles, cette approche devient encore plus intéressante. Néanmoins, la sélection des séquences homologues à utiliser pour la prédiction est un problème et il n'existe pas dans la littérature d'outil permettant d'effectuer cette sélection. La majorité des algorithmes de prédiction de structure secondaire d'ARN ne sont pas en mesure de prévoir une structure précise à partir d'un très grand alignement, et une seule séquence mal alignée peut "détruire" la prédiction. Ainsi, comme nous le montrons ci-dessous, étant donné un ensemble de séquences homologues, seulement quelques unes des combinaisons possibles de ces séquences donnent des prédictions correctes. Nous expliquons que cela peut être dû à la variabilité des séquences homologues et à la faible qualité de l'alignement dans les hélices.

Pour traiter le problème de l'alignement de séquences homologues, une approche consiste à réaliser en même temps la prédiction de la structure et l'alignement [165]. En raison de la très grande complexité d'une telle approche, très peu de séquences peuvent être prises en compte (en général 2 séquences). Une autre approche est de parvenir à une sélection du meilleur ensemble de séquences homologues à utiliser pour faire la prédiction de la structure. Dans les travaux publiés, aucune information n'est donnée sur la manière dont est réalisée cette sélection. Nous supposons que cela est fait manuellement. Nous avons pour notre part proposé un algorithme appelé *SSCA*, qui permet de faire ce choix automatiquement.

3.5.1 Pourquoi sélectionner les séquences homologues

Pour pouvoir utiliser une approche comparative pour la prédiction de structure secondaire d'ARN, il faut un ensemble de séquences homologues "bien alignées". En effet, au moins deux problèmes peuvent se produire avec les alignements de séquences d'ARNs lorsqu'on considère des contraintes de structure :

- Les séquences sont souvent très variables dans les zones qui sont situées à la périphérie de la structure. Ainsi dans ces régions, la comparaison des séquences est difficile ou impossible. Et inversement, le cœur de la structure est souvent moins variable, parfois très conservé, ne présentant donc pas assez d'information de covariation (mutations compensatoires). Par conséquent, pour arriver à avoir une bonne prédiction de structure par comparaison de séquences, il faut sélectionner des séquences homologues qui sont suffisamment différentes pour avoir des mutations compensatoires, mais assez proches pour être comparées.
- Les régions correspondant à des hélices semblent être plus variables que les régions simple brin [35, 81, 196]. Par conséquent, les alignements sont souvent de qualité médiocre dans ces régions, et les hélices sont souvent décalées. Ce point est crucial car, pour être efficace, l'approche comparative a besoin d'utiliser des séquences homologues avec des hélices correctement alignées et non décalées. Par conséquent, il faut trouver des critères pour différencier les séquences homologues avec des hélices décalées des séquences homologues qui sont correctement alignées.

Nous avons effectué les tests suivants pour démontrer comment la qualité de la prédiction peut varier selon les séquences homologues utilisées. Nous avons utilisé un alignement de 44 séquences d'ARNtm de la base de données tmRDB [191] et un alignement de 54 séquences de RNase P de la base de données RNase P [20]. Les séquences ont été choisies arbitrairement, en éliminant les séquences de moins de 30% d'identité et les séquences qui sont redondantes. Les séquences sont ré-alignées en utilisant le logiciel d'alignement multiple ClustalW [105] avant d'effectuer la prédiction de la structure, et ce afin d'éviter des informations sur la structure secondaire éventuellement déjà prises en compte dans l'alignement.

Nous avons utilisé notre algorithme *P-DCfold* pour prédire la structure secondaire de RNase P et de l'ARNtm de Escherichia coli. *P-DCfold* nécessite ici quatre séquences homologues. Nous avons ensuite effectué des prédictions en utilisant chaque combinaison de 4 séquences homologues à partir des alignements considérés. Nous avons calculé un score de qualité pour chaque prédiction après comparaison avec les structures de

référence connues fournies par les bases de données tmRDB et RNase P. Nous avons fixé un seuil de 0,75 au-dessus duquel nous avons considéré qu'une prédiction est bonne. Seulement quelques combinaisons possibles permettent de prédire correctement la structure : environ 1% pour l'ARNtm et le RNase P, comme on peut le voir dans la Table 3.2. Par conséquent, il y a seulement une chance sur cent d'obtenir une bonne prédiction sans critères de sélection des séquences homologues.

Une méthode courante pour réduire l'hétérogénéité des résultats de prédiction consiste à sélectionner des séquences homologues d'identité de séquences entre 60% et 80%, et d'éliminer les séquences identiques ou presque identiques à partir de l'alignement. C'est le modèle d'homologie commun M_{HC}, couramment utilisé. Nous avons utilisé cette méthode pour sélectionner dix séquences homologues à partir des alignements d'ARNtm et de RNase P considérés ci-dessus. Comme le montre la Table 3.2, les prédictions pour chaque combinaison de 4 séquences parmi ces 10 séquences ont de meilleurs scores de prédiction (MCC) que les prédictions utilisant toutes les séquences.

	Toutes les séquences		M_{HC}	
	ARNtm	RNaseP	ARNtm	RNaseP
Nombre total de prédictions	123410	266699	210	210
Nb de prédictions avec MCC>75	1620	1958	18	38
MCC moyen	45.19	41.03	56,82	60,27
MCC Maximal	89	86	85	84
MCC Minimal	10	5	26	30

TABLE 3.2 – Caractéristiques des prédictions de structure secondaire réalisées en utilisant *P-DCfold* sur un alignement de 44 séquences d'ARNtm et un alignement de 54 séquences de RNase P. A gauche : toutes les combinaisons possibles de quatre séquences homologues sont considérées. A droite : seules les combinaisons de quatre séquences entre 10 séquences homologues initialement sélectionnées par le modèle d'homologie M_{HC} sont considérées.

Ces résultats montrent l'importance du choix des séquences homologues pour prédire de manière efficace la structure secondaire de l'ARN. Le modèle M_{HC} améliore nettement les résultats de prédiction mais reste néanmoins insuffisant. Nous avons donc mis au point un algorithme pour sélectionner des combinaisons de séquences homologues qui donnent de meilleurs scores de prédiction (MCC) par rapport à ceux obtenus avec la méthode d'homologie commun M_{HC}.

3.5.2 Critères pour la sélection des séquences homologues

Les séquences homologues les plus appropriées sont celles qui ont une variabilité suffisante par rapport à la séquence cible, et un alignement correct au niveau des hélices. Cette information peut être évaluée à l'aide de matrices de substitution. Une matrice de substitution est construite pour chaque séquence homologue : elle contient tous les taux de substitution entre cette séquence et la séquence cible pour les quatre bases A, C, G et U.

Critère de variabilité

On veut sélectionner des séquences homologues qui sont suffisamment variables par rapport à la séquence cible pour présenter des mutations compensatoires mais assez proches pour être comparées. Nous avons défini la "variabilité suffisante" d'une séquence homologue par rapport à la séquence cible en fonction des pourcentages d'identités et de suppressions.

Une hélice est pertinente lorsque le nombre de substitutions compensatoires par base dépasse un seuil T. La probabilité de trouver des mutations compensatoires augmente avec le nombre de séquences utilisées.

Si N est le nombre de séquences homologues utilisées pour prédire la structure, le pourcentage adéquat d'identités I des séquences homologues est donc : $I = 1 - \frac{T}{N}$.

Le pourcentage de suppressions a été indexé sur le pourcentage d'identités. Cela nous permet d'éliminer les séquences homologues qui ont un pourcentage anormalement élevé de suppressions par rapport à leur pourcentage d'identités. Nous définissons le pourcentage adéquat D de suppressions comme suit : $D = \frac{I}{75}$.

Nous écartons également les séquences avec des bases ambiguës ou indéterminées.

Critère d'alignement au niveau des hélices

L'évolution agit de façon à conserver la structure d'une molécule d'ARN qui est essentielle pour sa fonction. La séquence d'une hélice est moins importante que l'appariement des bases. Ainsi les hélices présentent une variabilité dans leurs séquences beaucoup plus importante que dans les régions simple brin. Ceci implique que les régions simple brin sont généralement correctement alignées, alors que les régions d'hélices peuvent être mal alignées [35, 81].

Ce que nous suggérons ici est de définir des modèles permettant de différencier les séquence bien alignées au niveau des hélices de celles qui ne le sont pas. Pour cela, nous avons identifié des facteurs liés à la stabilité des hélices, à partir desquels nous avons ensuite défini les mutations qui sont "a priori" préférées par l'évolution :

- Stabilité des paires de base : La paire de bases GC est plus stable que la paire de bases AU qui est elle même plus stable que la paire de bases GU [51]. En raison de ces différences dans la stabilité, les paires de bases GC sont préférées lorsqu'une tige est importante pour le maintien de la structure globale, tandis que les paires de bases GU sont désavantagées. Le résultat est que les tiges sont constituées d'une majorité de paires de bases GC [79].
- Transitions versus Transversions : Les mutations qui comportent deux bases du même type (deux purines (A et G) ou de deux pyrimidines (C et U)) sont des transitions (G↔A et C↔U). Les autres sont des transversions (G↔U, C↔G, U↔A et C↔A). Les transitions se produisent plus facilement que les transversions [196]. Ce phénomène est accentué dans les tiges étant donné que les mutations comprennent des paires de bases.
- Stabilité des états intermédiaires : Les doubles mutations entre les paires de base ne peuvent pas apparaître simultanément en raison du faible taux de mutation, donc elles utilisent un état intermédiaire (par exemple AU→UU→UA). Les mutations doubles sont supportées ou désavantagées en fonction de la stabilité de cet état intermédiaire. Il peut être un état non apparié très délétère ou un état d'appariement GU qui n'est que légèrement délétère [163]. Néanmoins, l'état intermédiaire est rarement observé dans les alignements de séquences [55], et ce parce que l'état intermédiaire est maintenu rarement par la sélection [81]. Comme la paire GU est la plus stable et la moins nocive des états intermédiaires, les doubles substitutions qui l'utilisent peuvent se produire plus fréquemment que les autres [55].

Nous pouvons mesurer séparément l'influence de la stabilité de GC d'une part et l'influence de l'état intermédiaire GU avec l'influence des transitions/transversions d'autre part :

- L'influence de la stabilité de GC est mesurée en comparant A→C avec C→A, U→C avec C→U, A→G avec G→A et U→G avec G→U. L'idée ici est de favoriser les substitutions qui permettent l'apparition de G et C au profit de A et U.
- L'influence de l'état intermédiaire GU et l'influence des transitions/transversions sont mesurées en comparant les substitutions entre A→G et A→C, U→C et U→G, C→U et C→A, G→A et G→U. L'idée ici est de favoriser les substitutions qui permettent de passer par l'état intermédiaire tout en étant des transitions.

3.5.3 Algorithme

Notre algorithme pour la sélection des séquences homologues, appelé *SSCA* (Sequence Selection for the Comparative Approach) est donné en Figure 3.10.

Algorithme *SSCA* (S_t : séquence cible, A: Alignement de séquences homologues)

Début

- Construire un modèle $\mathcal{M}$
- Pour chaque séquence homologue S_i de A
  - Calculer la matrice substitution M_i entre S_t and S_i
  - Calculer le score pour S_i en fonction des contraintes du modèle $\mathcal{M}$ et de la matrice de substitution M_i
- Trier les séquences S_i selon leur score
- Sélectionner le nombre adéquat de séquences pour prédire la structure de S_t

End

FIGURE 3.10 – Algorithme *SSCA* pour la sélection des séquences homologues

SSCA prend en entrée une séquence cible et un ensemble de séquences alignées homologues. Il est basé sur l'utilisation d'un modèle de sélection, appelé modèle $\mathcal{M}$, qui consiste en des contraintes sur les matrices de substitution des séquences homologues par rapport à la séquence cible. Ces contraintes modélisent la séquence homologue idéale ayant une variabilité suffisante et un alignement correct des hélices. Pour chaque séquence homologue, *SSCA* calcule d'abord la matrice de substitution entre cette séquence et la séquence cible (voir Figure 3.11). Un score d'intérêt est ensuite calculé pour la séquence en fonction du modèle et de sa matrice de substitution. Les séquences les plus proches de la séquence idéale (modélisée par $\mathcal{M}$) sont les plus intéressantes pour la prédiction de la structure.

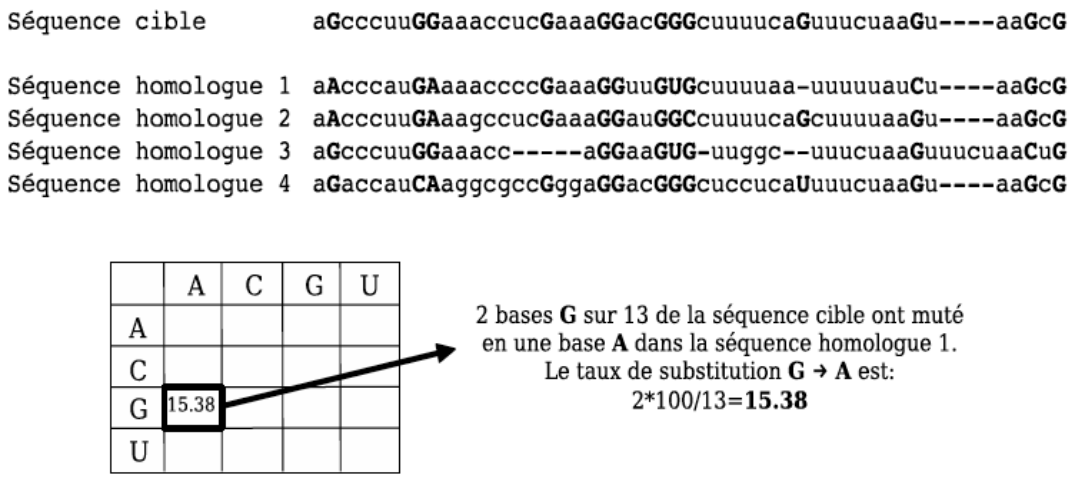

FIGURE 3.11 – Calcul de la matrice de substitution entre une séquence homologue (ici la première dans l'alignement) et la séquence cible

Le modèle $\mathcal{M}$ est composé de deux parties : l'une concerne les contraintes de variabilité de séquence et l'autre concerne les contraintes d'alignement au niveau des hélices.

Contraintes sur la variabilité des séquences Les contraintes pour la sélection des séquences homologues en fonction de leur variabilité sont :

$$\begin{cases} C1{=}|(x \to x) - I| & x \in \{A, C, G, U\} \\ C2{=}|(x \to' -') - D| & x \in \{A, C, G, U\} \\ C3{=}|(x \to' N')| & x \in \{A, C, G, U\} \\ S_H{=}C1 + C2 + C3 \end{cases}$$

La contrainte $C1$ favorise les séquences homologues avec un taux d'identité proche de celui décrit dans l'équation $I = 1 - \frac{T}{N}$. La contrainte $C2$ favorise les séquences homologues avec un taux de suppressions proche de celui décrit dans l'équation $D = \frac{I}{75}$, tandis que $C3$ favorise les séquences sans ambiguïté et sans bases indéterminées ('N'). Les séquence homologues avec une variabilité adéquate sont ainsi sélectionnées en minimisant la somme S_H des contraintes $C1$, $C2$ et $C3$.

Contraintes sur les alignements d'hélices Il existe deux méthodes pour la construction de la deuxième partie du modèle lié à l'alignement dans les hélices. Ces méthodes mettent l'accent sur les trois influences dans les régions hélices décrites dans la Section 3.5.2. Chaque méthode fournit un modèle qui peut être utilisé dans notre algorithme *SSCA*.

– La première méthode consiste à mesurer l'influence de l'état intermédiaire GU et les différences entre les transitions et transversions. A→G est comparé à A→C, U→C à U→G, C→U à C→A et G→A à G→U, en appliquant les contraintes suivantes sur les cases de la matrice de substitution pour chaque séquence homologue:

$$\begin{cases} C4 = (A \to G) - (A \to C) \\ C5 = (U \to C) - (U \to G) \\ C6 = (C \to U) - (C \to A) \\ C7 = (G \to A) - (G \to U) \\ S_{A1}{=}C4 + C5 + C6 + C7 \end{cases}$$

Les contraintes C4, C5, C6 et C7 mesurent les différences entre les taux de substitution A→G, U→C, C→U et G→A et les taux de substitution A→C, U→G, C→A et G→U. Le score S_{A1} est ainsi maximisé pour sélectionner les séquences qui sont largement influencées par l'état intermédiaire GU.

– La seconde méthode mesure la stabilité GC. Elle compare A→C avec C→A, U→C avec C→U, A→G avec G→A et U→G avec G→U, en utilisant les contraintes suivantes :

$$\begin{cases} C4 = (A \to C) - (C \to A) \\ C5 = (A \to G) - (G \to A) \\ C6 = (U \to C) - (C \to U) \\ C7 = (U \to G) - (G \to U) \\ S_{A2}{=}C4 + C5 + C6 + C7 \end{cases}$$

Les contraintes C4, C5, C6 et C7 mesurent les différences entre les taux de substitution A→C, A→G, U→C et U→G et les taux de substitution C→A, G→A, C→U et G→U. Le score S_{A2} est ainsi maximisé pour sélectionner les séquences qui sont largement influencées par l'état stable GC.

Modèles pour la sélection des séquences homologues Chaque méthode de calcul de la deuxième partie du modèle (alignement d'hélice) est combinée avec la méthode de calcul de la première partie du modèle (la variabilité). Nous avons ainsi obtenu deux modèles pour la sélection des séquences homologues, qui peuvent chacun être utilisé pour calculer un score pour chaque séquence homologue :

– modèle $\mathcal{M}_{\mathcal{GU}}$ avec un score $S_{GU} = S_H - S_{A1}$
– modèle $\mathcal{M}_{\mathcal{GC}}$ avec un score $S_{GC} = S_H - S_{A2}$

Un autre modèle, qui est une combinaison des deux modèles ci-dessus, fournit une mesure de l'influence combinée de la stabilité GC et de l'état intermédiaire GU :

– modèle $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ avec un score de $S_{GC+GU} = S_{GC} + S_{GU}$

Les séquences homologues à utiliser pour prédire la structure de la séquence cible sont sélectionnées en fonction de leur scores S_{GU}, S_{GC} ou S_{GC+GU}. Les séquences homologues les plus appropriées pour l'approche comparative ont les scores les plus faibles (puisque S_{A1} et S_{A2} sont maximisés et S_H minimisé). L'algorithme *SSCA* a été ainsi testé en utilisant chacun des trois modèles $\mathcal{M}_{\mathcal{GU}}$, $\mathcal{M}_{\mathcal{GC}}$ et $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$.

3.5.4 Résultats

SSCA a été testé sur plusieurs alignements de séquences d'ARNs, dont l'ARNtm et le RNaseP. Nous avons utilisé *P-DCfold* pour prédire la structure secondaire d'une séquence cible donnée. Les séquences de ces deux ARNs ayant des tailles avoisinant les 380 nucléotides, le nombre de séquences homologues nécessaires à *P-DCfold* pour prédire la structure secondaire de la séquence cible est de 4 séquences.

Un alignement de 44 sequences et une structure de référence provenant de la Base de données tmRDB [220] et un alignement of 54 séquences et une structure de référence provenant de la Base de données RNaseP [20] ont été utilisés pour la prédiction de la structure secondaire de *Escherichia coli* ARNtm et de *Escherichia coli* RNAse P respectivement.

Afin de tester et comparer les trois modèles de *SSCA*, nous avons procédé comme suit :

1. Nous avons prédit la structure de la séquence cible avec *P-DCfold* en utilisant toutes les combinaisons possibles des quatre séquences homologues. Puis nous avons calculé et attribué les scores MCC à chaque prédiction obtenue.
2. L'algorithme *SSCA* a été utilisé pour classifier les séquences homologues en fonction des scores obtenus avec chaque modèle $\mathcal{M}_{\mathcal{GU}}$, $\mathcal{M}_{\mathcal{GC}}$ et $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$. Nous les avons également classifiées en fonction du modèle commun d'homologie M_{HC}.
3. Les dix meilleures séquences homologues pour chaque classification ont été sélectionnées et chaque combinaison possible des quatre séquences homologues ont été testées. Nous avons donc effectué 210 prédictions et les score MCC de chacune d'elles a été calculé.

Les résultats obtenus avec les quatre modèles M_{HC}, $\mathcal{M}_{\mathcal{GU}}$, $\mathcal{M}_{\mathcal{GC}}$ et $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ pour l'ARNtm et le RNAse P sont donnés en Table 3.3.

	ARNtm					RNAse P				
	Tous	M_{HC}	$\mathcal{M}_{\mathcal{GU}}$	$\mathcal{M}_{\mathcal{GC}}$	$\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$	Tous	M_{HC}	$\mathcal{M}_{\mathcal{GU}}$	$\mathcal{M}_{\mathcal{GC}}$	$\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$
Moy MCC	45.19	56.82	63.38	67.66	67.45	41.03	60.27	73.58	70.13	75.3
Max MCC	89	85	84	80	85	86	84	85	80	85
Min MCC	10	26	41	56	41	5	30	56	56	56
%MCC$>$ 75	1.3%	8.6%	5.7%	27.6%	26.7%	0.7%	18%	48.6%	23.3%	60.4%

TABLE 3.3 – Distribution du MCC des prédictions de structure secondaire de l'ARNtm et de RNase P réalisées avec l'algorithme *P-DCfold* , et en utilisant différents modèles de sélection de séquences homologues.

Comme on peut le voir, les trois modèles $\mathcal{M}_{\mathcal{GU}}$, $\mathcal{M}_{\mathcal{GC}}$ et $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ que nous avons définis donnent de meilleurs scores MCC que ceux obtenus en utilisant toutes les séquences, et meilleurs que ceux obtenus en utilisant le modèle classique M_{HC} pour les deux ARNs. Dans le cas de l'ARNtm, il y a environ une chance sur 4 de prédire la bonne structure avec le modèle $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ alors qu'il n' y a qu'une chance sur 100 si on utilise toutes les séquences, et seulement une chance sur 10 environ en considérant le modèle M_{HC}. Dans le cas de RNaseP, les résultats sont encore plus probants. ll y a plus d'une chance sur 2 de prédire la bonne structure avec le modèle $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ alors qu'il y a moins d'une chance sur 100 si on utilise toutes les séquences, et moins d'une chance sur 5 en considérant le modèle M_{HC}.

Nous avons vérifié la capacité du modèle $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$ à mesurer l'utilité de séquences données pour l'approche comparative. Nous avons utilisé les deux alignements de ARNtm et de RNase P et prédit les structures secondaires des séquences d'Escherichia Coli pour toutes les combinaisons possibles de quatre séquences homologues, en utilisant l'algorithme *P-DCfold* . Nous avons calculé d'une part le score moyen MCC et d'autre part le score *SSCA* pour chaque séquence homologue. Plus le score de *SSCA* est bas, plus la séquence a une forte probabilité d'être appropriée pour être utilisée comme séquence homologue dans la prédiction. Et inversement, les séquences les plus utiles pour la prédiction de la structure secondaire ont les scores en MCC moyen les plus élevés. La Figure 3.12 donne la corrélation entre le score moyen MCC de chaque séquence homologue et le score attribué à cette séquence avec le modèle $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$. Comme on peut le voir, les séquences homologues avec les scores *SSCA* les plus bas ont les plus hauts scores moyens MCC. Ceci valide ainsi notre modèle et algorithme de sélection de séquences homologues.

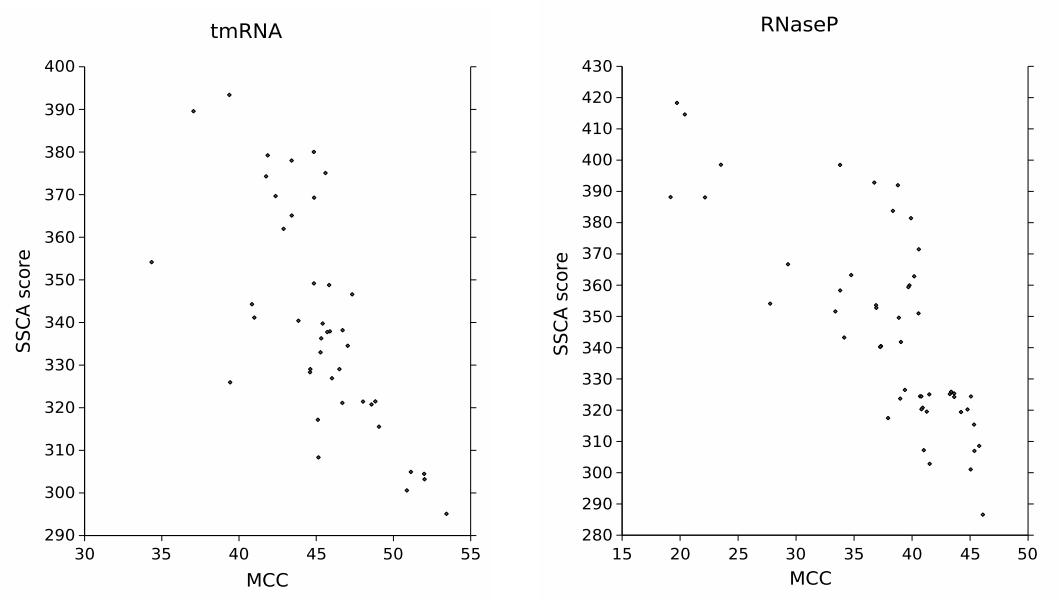

FIGURE 3.12 – Corrélation entre les scores de *SSCA* ($\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$) et les scores MCC moyens obtenus sur des alignements de séquences homologues de ARNtm (à gauche) et de RNase P (à droite). Les séquences homologues avec les scores *SSCA* les plus bas ont les scores les plus élevés en MCC moyen.

3.5.5 Conclusion

Nous avons développé un algorithme, appelé *SSCA*, pour sélectionner des séquences homologues pour une utilisation dans la prédiction de la structure secondaire d'ARN basée sur l'approche comparative. La sélection des séquences homologues est basée sur l'idée que les contraintes de structure biaisent les matrices de substitution dans les hélices. Nous avons défini trois modèles de sélection : $\mathcal{M}_{\mathcal{GU}}$, basé sur des contraintes de l'état intermédiaire GU ; $\mathcal{M}_{\mathcal{GC}}$, basé sur des contraintes de stabilité de GC ; et $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$, basé à la fois sur la stabilité des GC et des contraintes de l'état intermédiaire GU.

Nous avons comparé nos trois modèles avec un modèle actuellement utilisé, le modèle M_{HC}, en prédisant les structures secondaires de l'ARNtm et de RNase P avec l'algorithme *P-DCfold*. Les trois modèles améliorent de manière significative la probabilité d'obtenir de bonnes prédictions et ont donné de meilleurs résultats que le modèle M_{HC}. Le meilleur modèle est $\mathcal{M}_{\mathcal{GC}+\mathcal{GU}}$, qui a trois fois plus de bonnes prédictions que le modèle M_{HC}.

La complexité en temps de l'algorithme *SSCA* est $O(m \times n)$, avec n la longueur de la séquence cible et m le nombre de séquences homologues. Tous nos tests ont été effectués en moins de 5 secondes.

3.6 Tfold : Algorithme efficace pour la prédiction de structure secondaire d'ARN incluant les pseudonœuds

SSCA a été intégré à *P-DCfold* afin d'obtenir un algorithme de prédiction de structure secondaire efficace, basé sur une approche comparative et non dépendant des séquences homologues données en entrée et de leur alignement. L'algorithme obtenu est appelé *Tfold*. *Tfold* utilise la même approche algorithmique globale que *P-DCfold*, mais comme nous le montrons ci-dessous, présente un nombre important d'améliorations et d'extensions par rapport à ce dernier, en particulier au niveau de la sélection des hélices.

3.6.1 Principe et Algorithme

Procédure principale

L'algorithme *Tfold*, donné en Figure 3.13, prend en entrée un ensemble de séquences alignées, sélectionne les séquences les mieux alignées et les plus pertinentes par rapport à la séquence cible choisie, puis lance la prédiction plusieurs fois avec des sous-ensembles de séquences tests différents. Les hélices prédites au moins par la moitié des prédictions sont ainsi sélectionnées, pour former la structure finale.

Algorithme *Tfold* (S : séquence cible, A: Alignement de séquences homologues)
Début
$J \leftarrow \emptyset$
$E \leftarrow \emptyset$
  $H \leftarrow$ *Sélection_Séquences* (S, A)
  Pour chaque combinaison J_K de N_t séquences parmi les séquences de H
      $J \leftarrow J \cup J_K$
      $E_k \leftarrow$ *Recherche_Toutes_hélices* (S, J_K)
      $E \leftarrow E \cup E_k$
    fin pour
  $SS \leftarrow$ *Prédiction_Commune* (E, J)
  retourner SS
Fin

FIGURE 3.13 – Algorithme *Tfold*

Tfold prend en entrée une séquence cible S, pour laquelle une structure secondaire est recherchée, et un alignement A de séquences homologues. La première étape de *Tfold* est d'obtenir à partir des séquences de A un sous-ensemble des meilleures séquences à utiliser pour la prédiction de la structure secondaire SS de S. Il s'agit de la procédure *Sélection_Séquences*, qui lance *SSCA*, puis qui récupère un sous-ensemble H de séquences (par défaut 10 séquences) avec les scores les plus faibles et donc représentant les séquences les plus informatives pour la prédiction de la structure secondaire de S. Ensuite, pour chaque combinaison J_K de N_t séquences parmi ces séquences, une structure secondaire est prédite pour la séquence cible (procédure *Recherche_Toutes_hélices*). N_t représente le nombre de séquences homologues dont la procédure *Recherche_Toutes_hélices* a besoin pour la recherche d'hélices dans la séquence cible S. Ce paramètre est défini par l'utilisateur et est par défaut égal à 4. Enfin, la dernière étape de *Tfold* consiste à sélectionner les hélices qui sont retournées dans un nombre minimal de prédictions, afin d'obtenir la structure optimale (procédure *Prédiction_Commune*). Nous décrivons ci-après les différentes étapes de notre algorithme.

Prédiction commune

Le principe de cette méthode est la suivante: Soit K ensembles notés J_k ($k \in [1; K]$) de séquences homologues tel que chaque ensemble est utilisé pour prédire la structure secondaire d'une même séquence cible. Chaque prédiction retourne un ensemble d'hélices. Pour chaque hélice H_j ($j \geq 1$) apparaissant dans au moins une structure est associé un nombre (non nul) d'occurrences A_{H_j} dans les différentes structures.

Une hélice peut être présente sous des formes "équivalentes". Nous définissons ci-dessous la notion d'équivalence entre hélices. Mais avant, nous avons besoin de définir la relation de "sous-hélice", notée R_{ss}, entre deux hélices :

Definition (Relation de sous-hélice) : *Soit deux hélices H_1 et H_2 définies respectivement par (b_1,e_1,l_1) et (b_2,e_2,l_2), où b_i, e_i et l_i sont respectivement la position du premier brin, la position du second brin et la longueur de H_i. H_1 est une sous-hélice de H_2 ($H_1 R_{ss} H_2$) s'il existe un entier $d \geq 0$ tel que :*

$$\begin{cases} b_1 - b_2 = e_1 - e_2 = d \\ (b_1 + l_1) - (b_2 + l_2) = (e_2 - l_2) - (e_1 - l_1) \end{cases}$$

Ainsi, nous pouvons déduire la relation d'équivalence R_{eq} entre deux hélices :

Definition (Relation d'équivalence entre deux hélice) : *Deux hélices H_1 et H_2 sont équivalentes ($H_1 R_{eq} H_2$) si H_1 R_{ss} H_2 ou H_2 R_{ss} H_1*

La structure commune sera composée des hélices ayant un nombre maximal d'occurrences. Une hélice H_j est sélectionnée si $A_{H_j} > K/2$. Si le nombre de séquences est élevé (plus grand ou égal à 100), ce seuil est fixé à $3K/4$.

Recherche des hélices de la structure

Plusieurs améliorations et extensions ont été apportées à la recherche des hélices dans *Tfold* par rapport à *DCfold* et *P-DCfold* :

- La recherche des hélices sur la séquence cible se fait grâce à une matrice où des scores thermodynamiques sont associés aux différents appariements.
- Au niveau de chaque séquence test, la vérification de la conservation de l'hélice se fait en alignant les deux sous-séquences associées, permettant ainsi de tenir compte des hélices avec erreurs.
- Plusieurs solutions de structures secondaires possibles peuvent être proposées pour un ARN donné.
- L'utilisateur peut renseigner une ou plusieurs hélices de la structure.

Recherche des hélices dans la séquence cible Les hélices sont recherchées dans la séquence cible en utilisant des critères de longueur combinés à des critères thermodynamiques. Seules les hélices suffisamment longues et qui satisfont les règles thermodynamiques de stabilité sont sélectionnées. Le seuil de longueur $lmin$ est le même que dans *DCfold* et *P-DCfold*, à savoir $log_4(n)$, où n est la longueur de la séquence. Les règles thermodynamiques que nous utilisons sont résumées dans la Figure 3.14. Ces règles sont liées à :

- Type des appariements de l'hélice. Les appariements GC sont plus stables que les appariements AU, qui à leur tour sont plus stables que les appariements GU [79]. En outre, la position des paires GU dans l'hélice a un effet sur la stabilité de l'hélice [55].
- Type des appariements délimitant le début et/ou la fin d'une hélice : lorsque l'appariement est AG ou AA, l'hélice est plus stable [44].
- Taille et type des boucles terminales : une hélice est plus stable lorsque la boucle formée est une tétraboucle de Type GNRA, UNCG ou CUYG (N : n'importe quelle base, R : purine et Y : pyrimidine) [205].

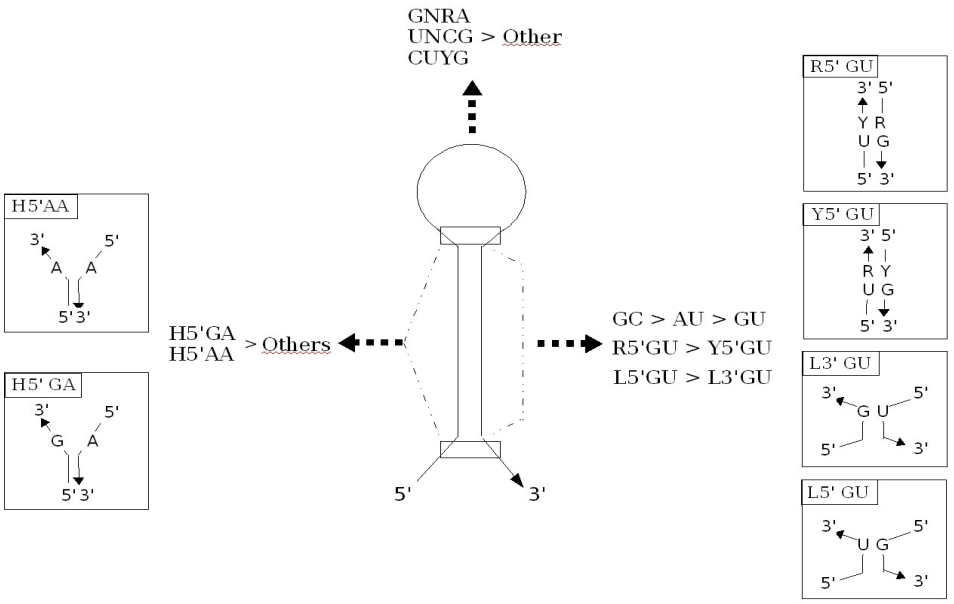

FIGURE 3.14 – Règles de stabilité du modèle d'hélice. A gauche : les règles concernant les mi-appariements aux extrémités de l'hélice. En haut : les règles concernant les boucles. A droite : les règles concernant les appariements GC, AU et GU dans l'hélice ainsi que la position de l'appariement GU (N : n'importe quelle base ; R : purine et Y : pyrimidine ; L : pour une boucle et H : pour une hélice).

Tfold utilise une matrice M de longueur (n, n) où la séquence cible S_t, de longueur n, est comparée à elle-même en sens inversé (voir Figure 3.15). Notons qu'il s'agit ici d'une matrice symétrique.

Pour chaque i, j de 1 à n :

$$M[i,j] = \begin{cases} M[i-1, j-1] + s(i,j) & \text{si } s(i,j) > 0 \\ 0 & \text{sinon} \end{cases}$$

où $s(i,j)$ est le score attribué à l'appariement $M[i,j]$ et dépendant de l'appariement ($S_t(i)$, $S_t(n+1-j)$). Il est égal à : 3 s'il s'agit d'un appariement GC ; 2 si c'est un appariement AU ou GU dans la configuration R5'GU ou L5'GU (R : pour purine, L : pour une boucle) ; 1 si c'est un appariement GU dans la configuration L3'GU ou Y5'GU (Y pour pyrimidine) ; 0 sinon.

Le score obtenu $s(i,j)$, lorsqu'il est plus grand que 0, est augmenté de 1 lorsque :

- $s(i-1, j-1) = 0$ et ($S_t(i-1)$, $S_t(n+1-(j-1))$) forme un appariement AG ou AA dans la configuration H5'GA ou H5'AA (H: pour hélice).

- $s(i+1, j+1) = 0$ et ($S_t(i+1)$, $S_t(n+1-(j+1))$) forme un appariement AG or AA dans la configuration H5'GA ou H5'AA.

Il est également augmenté de 2 lorsque :

- $s(i+1, j+1) = 0$ et l'hélice obtenue forme une boucle de taille 4 (tetra-loop) de la forme : GNRA, UNCG ou CUYG.

Le score du dernier appariement de l'hélice définit le score global de l'hélice.

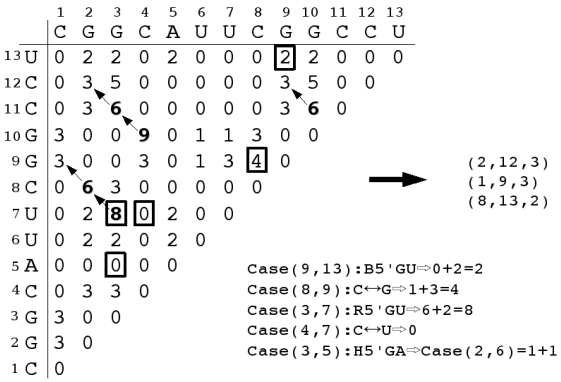

FIGURE 3.15 – Exemple de matrice d'appariement utilisée pour la recherche des hélices dans la séquence cible, avec prise en compte des critères thermodynamiques. Des scores d'appariement différents ont été attribués selon le type d'appariement : +3 pour GC, +2 pour AU, +2 pour GU en configuration B5'GU et R5'GU, +1 pour GU en configuration B3'GU et Y5'GU, +1 aux appariements fermants lorsqu'ils sont suivis d'une opposition AA ou AG ou lorsqu'ils sont suivis d'une tetraboucle. Les cases en gras correspondent aux cases sélectionnées pour $lmin = 3$.

Vérification de la conservation des hélices dans les séquences homologues Une fois les hélices déterminées dans la séquence cible S_t, on vérifie leur conservation dans les séquences homologues. Le principe est le suivant:

Soient b, e et l respectivement la position du premier brin, la position du second brin et la longueur de l'hélice sélectionnée dans la séquence S_t. Pour chaque séquence homologue S_H, nous considérons la sous-séquence $S_H[b-d, b+l+d]$ et la sous-séquence inverse de $S_H[e-d, e+l+d]$ où d représente un décalage possible ; ces deux séquences sont comparées et alignées en utilisant une matrice de score A de longueur $(l+2d, l+2d)$ (méthode de programmation dynamique). Ainsi, les hélices avec des renflements et des boucles internes sont prises en compte. Nous avons établi un score de -1 pour une insertion ou une suppression et -2 pour une paire de bases qui n'est pas un appariement GC, AU ou GU. Pour les appariements GC, AU et GU, nous avons considéré les mêmes scores que ceux donnés plus haut : +3 pour GC, +2 pour AU et pour GU dans les configurations R5'GU ou L5'GU, et +1 pour GU dans les configurations L3'GU ou Y5'GU. Le score global est ensuite augmenté dans le cas de boucles particulières et/ou d'hélices particulières, comme décrit dans la Figure 3.14. On en déduit ensuite un score de conservation de l'hélice dans la séquence S_H: il est égal au meilleur score dans la matrice A.

Prédiction de plusieurs structures alternatives Une étape importante dans notre algorithme est de vérifier la compatibilité des hélices sélectionnées (hélices conservées dans l'ensemble des séquences homologues avec des scores suffisamment élevés) et de traiter les éventuelles incompatibilités. La compatibilité entre toutes les hélices est une condition importante pour la subdivision de la séquence afin de rechercher d'autres hélices (voir Section 3.3.4). Lorsque deux hélices sont incompatibles, un choix doit être effectué, seule l'une des deux devant être gardée. Dans la procédure de traitement de compatibilité définie dans *DCfold*, lorsque deux hélices sont incompatibles, nous gardons celle ayant le score le plus élevé, et dans le cas où elles ont le même score ou des scores équivalents (et sont donc en conflit), les deux sont éliminées.

En effet, les critères de sélection que nous utilisons ne permettent pas toujours ou difficilement de faire un choix entre deux hélices potentielles. Dans *Tfold*, nous avons donc mis en place une nouvelle procédure qui traite d'une autre manière les conflits entre les hélices : nous gardons les deux hélices et proposons deux ensembles de points d'ancrage, chaque ensemble permettant de subdiviser la séquence en deux manières différentes, puis de chercher deux nouveaux ensembles d'hélices. Par conséquent, $k+1$ structures alternatives sont prédites avec k le nombre de conflits.

Deux hélices incompatibles sont considérées en conflit lorsqu'elles ont des scores proches. La notion de "scores proches" est définie par un paramètre dont la valeur peut être fixée par l'utilisateur. Par défaut, il est égal à zéro, ce qui signifie dans ce cas que deux hélices sont considérées en conflit seulement quand elles ont des scores égaux. Bien sûr, plus grande est la valeur du paramètre plus grand est le nombre de structures alternatives.

Prise en compte d'hélices pré-définies par l'utilisateur Il est fréquent qu'un biologiste utilisant un outil de prédiction de structure secondaire d'ARN connaît déjà une ou plusieurs hélices de la structure. Par conséquent, il aimerait avoir la possibilité de fixer ces hélices, et de pouvoir prédire grâce à un outil les autres hélices de la structure. Or à notre connaissance il n'existe pas de logiciel qui permet cela. Nous avons pour notre part intégré dans *Tfold* la possibilité de prise en compte d'hélices pré-définies.

Les hélices définies par l'utilisateur sont considérées comme des points d'ancrage dans *Tfold*. Elles sont insérées dans le premier ensemble de points d'ancrage et se voient attribuer un score maximal. Ainsi, elles sont toujours maintenues dans les différentes étapes de l'algorithme et la sélection des autres hélices est réalisée en tenant compte de ces hélices.

3.6.2 Résultats

Dans cette section, nous présentons les résultats que nous avons obtenus avec *Tfold* sur plusieurs ARNs non codants. Les résultats sont comparés à différents algorithmes et logiciels de l'état de l'art du domaine.

Deux analyses comparatives ont été réalisées : la première analyse a concerné la comparaison de *Tfold* avec des logiciels ne recherchant pas les pseudonœuds, et la seconde analyse a concerné la comparaison de *Tfold* avec des logiciels recherchant les pseudonœuds.

Nos tests ont été effectués sur plusieurs ARNs non-codants, de différents tailles et contenant ou non des pseudonœuds. Ces ARNs sont les suivants (donnés ici par ordre croissant de la longueur) : ARNt, l'ARN 5S, l'ARN U1, l'ARN SRP, ARNtm, RNase P, l'ARN 16S et l'ARN 23S. Pour chaque ARN, un alignement de séquences homologues a été récupéré à partir d'une base de données. Les séquences ont été ré-alignées en utilisant ClustalW [105], afin d'éviter toute information sur la structure secondaire dans l'alignement. Pour chaque ARN la prédiction de la structure secondaire a été effectuée pour une séquence (sélectionnée à partir de l'alignement), considérée comme la séquence cible. Les différentes séquences d'ARN utilisées pour nos tests sont disponibles sur notre serveur web (`http://EvryRNA.ibisc.univ-evry.fr`) (voir Chapitre 6).

Résultats de Tfold sans prise en compte des pseudonœuds

Dans une première analyse, nous avons comparé notre algorithme *Tfold* avec plusieurs logiciels existants pour la prédiction de structure secondaire d'ARN sans pseudonœuds : *Mfold* [126, 129], *RNAalifold* , [79, 160], *Pfold* [97, 149], *Carnac* [148, 23] et *LocARNA* [204]. Ces logiciels sont de différents types : *Mfold* effectue la prédiction de la structure secondaire d'une seule séquence, tandis que *Pfold*, *Carnac* et *RNAalifold* prédisent la structure secondaire commune de plusieurs séquences alignées homologues (deux dans le cas de *Carnac*), sachant que *Pfold* et *Carnac* renvoient aussi la structure secondaire de chaque séquence, ce qui n'est pas le cas de *RNAalifold*.

La Figure 3.16 donne la sensibilité et la sélectivité (selon la définition de Gardner et Giegerich [54], voir Section 3.2.2) obtenues par chacun des logiciels considérés sur les différents ARNs et la Figure 3.17 donne les résultats de corrélation.

Pour chaque ARN considéré, les résultats *Tfold* en terme de sensibilité, de sélectivité et de MCC sont toujours parmi les deux meilleurs résultats (sauf pour l'ARN U1 où la sélectivité et le MCC sont les troi-

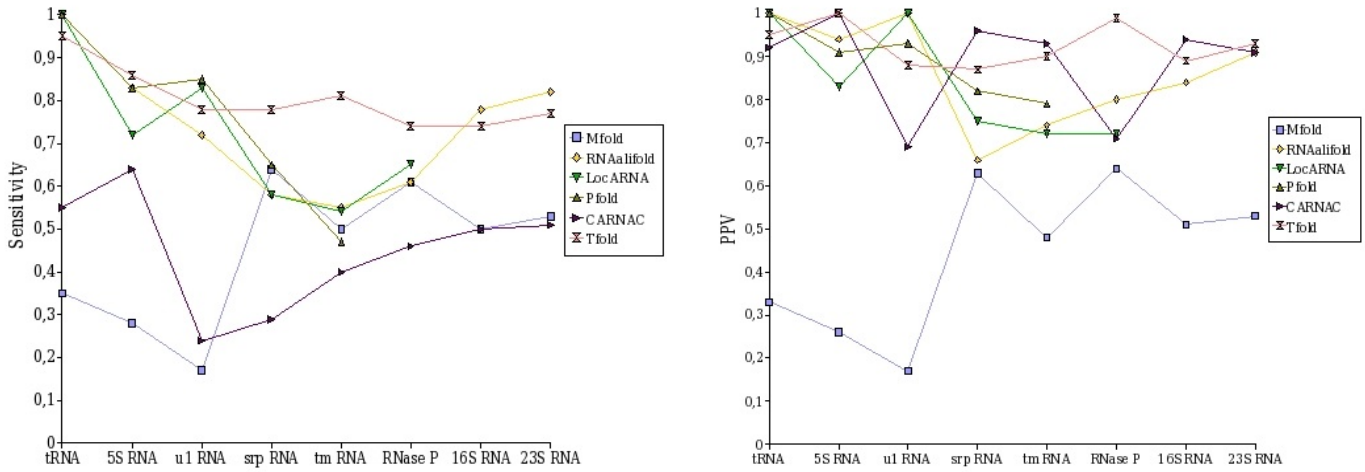

FIGURE 3.16 – Les résultats obtenus par Tfold et par d'autres logiciels de prédiction de structures secondaires d'ARN. Gauche : Résultats de sensibilité. Droite : Résultats de sélectivité.

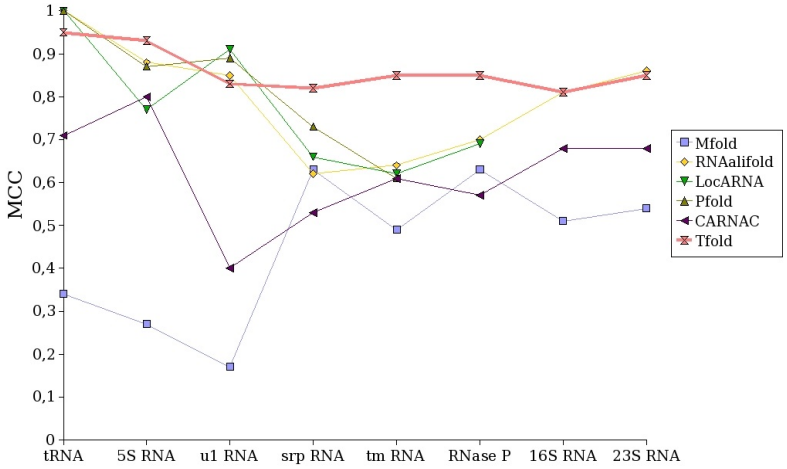

FIGURE 3.17 – Résultats de corrélation (MCC) obtenus par Tfold et par d'autres logiciels de prédiction de structures secondaires d'ARN.

sièmes meilleurs résultats), comme nous pouvons le voir dans la Figure 3.16 et la Figure 3.17. La sensibilité moyenne est approximativement de 0,8, ce qui signifie que 80% des appariements de la structure secondaire sont trouvés, et la sélectivité se situe autour de 0,90, ce qui signifie que 10% seulement des appariements prédits sont des faux positifs. En outre, la corrélation (moyenne de 0,85) est très bonne par rapport aux autres logiciels. *Tfold* est le seul logiciel qui obtient une corrélation toujours supérieure à 0,80. Enfin, le point important à souligner est que les résultats obtenus par *Tfold* sont homogènes pour tout ARN considéré, et donc quelque soit sa taille, contrairement aux autres logiciels. En l'occurrence, *Pfold* n'a pas pu prédire les structures de RNase P, de l'ARN 16S et de l'ARN 23S et *LocARNA* n'a pas pu prédire celles de l'ARN 16S et 23S.

Résultats sur la prédiction de pseudonœuds

L'une des grandes forces de Tfold est sa capacité à prédire les pseudonœuds. Pour évaluer son efficacité sur cet aspect, nous avons voulu le comparer aux algorithmes et méthodes existants dans la littérature et

spécialisés dans la prédiction de pseudonœuds, à savoir ILM [164], *vsfold* [38, 198], *pknotsRG* [158, 154], *knotseeker* [174], *HFold* [84] et *IP-based method* [155]. Parmi ces travaux, seuls quatre fournissaient le programme associé ou un serveur web permettant d'effectuer des comparaisons en terme de résultats de prédiction. Il s'agit de ILM, pknotsRG, vsfold et knotseeker.

La Figure 3.18 montre les résultats en termes de sensibilité et de sélectivité obtenus par Tfold et chacun des logiciels ILM, pknotsRG et vsfold dans la prédiction de la structure secondaire incluant les pseudonœuds de plusieurs ARNs, et la Figure 3.19 montre les résultats de corrélation. Le logiciel knotseeker n'a pas pu être considéré ici car il ne replie pas toute la séquence, mais recherche seulement les pseudonœuds. Nous donnons ses résultats dans la Table 3.4.

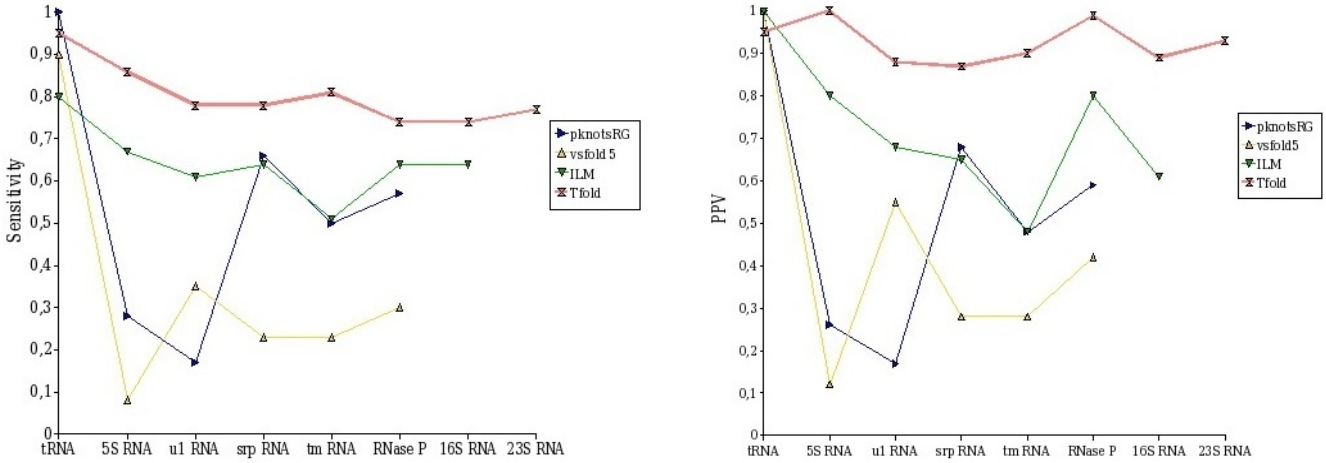

FIGURE 3.18 – Résultats obtenus par *Tfold* et plusieurs autres logiciels de prédiction de structure secondaire d'ARN incluant les pseudonœuds. Gauche : Résultats de sensibilité. Droite : Résultats de sélectivité.

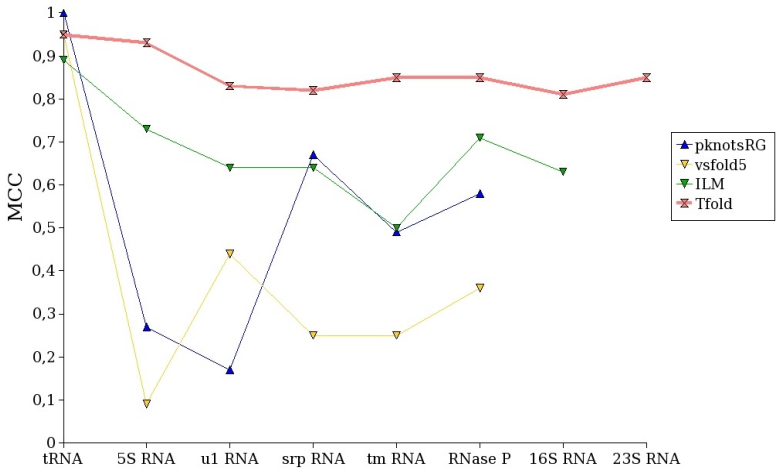

FIGURE 3.19 – Résultats de corrélation (MCC) obtenus par *Tfold* et plusieurs autres logiciels de prédiction de structure secondaire d'ARN incluant les pseudonœuds.

Comme nous pouvons le voir, *Tfold* donne de meilleurs résultats en comparaison aux autres logiciels et ce sur tous les ARNs considérés, à l'exception de l'ARNt. Nous pouvons aussi constater que Tfold est le seul à pouvoir prédire la structure avec ses pseudonœuds de toutes les séquences quelque soit leur taille, y compris pour l'ARN 16S et 23S. ILM n'a rien retourné pour le 23S, et vsfold et pknotsRG n'ont rien pu retourner et pour le 16S et pour le 23S. En effet, ces outils, en particulier knotseeker, vsfold et pknotsRG, sont plus dédiés

au traitement de petites séquences, de longueur entre 21 et 137 pb, correspondant à des pseudonœuds. Pour ne pas pénaliser ces programmes qui sont donc plus destinés à la recherche de pseudonœuds, nous avons regardé plus précisément les pseudonœuds trouvés pour chacun des ARNs considérés. Les résultats sont données dans la Table 3.4.

	ARNt	ARN 5S	ARN u1	ARN SRP	ARNtm	RNAseP
knotSeeker	0/1 1 FP	0/0 0 FP	0/0 1 FP	0/0 1 FP	2/4 1 FP	0/2 1 FP
pknotsRG	0/1 0 FP	0/0 0 FP	0/0 0 FP	0/0 1 FP	0/4 0 FP	0/2 0 FP
vsfold5	0/1 1 FP	0/0 1 FP	0/0 2 FP	1/1 5 FP	1/4 2 FP	0/2 4 FP
Tfold	0/1 0 FP	0/0 0 FP	0/0 0 FP	1/1 0 FP	3/4 1FP	2/2 0 FP

TABLE 3.4 – Résultas obtenus par knotSeeker, pknotsRG, vsfold et Tfold pour la prédiction des pseudonœuds apparaissant dans les ARNs suivants : ARNt, ARN 5S, ARN u1, ARN SRP, ARNtm et RNAseP. Les résultats sont donnés en terme de : (i) nombre de pseudonœuds prédits sur le nombre total de pseudonœuds connus ; et (ii) nombre de pseudonœuds faux positifs prédits (FP).

Comme on peut le constater avec les résultats donnés dans la Table 3.4, Tfold a l'avantage de prédire très peu de pseudonœuds faux positifs (seulement 1 dans le cas de l'ARNtm) et de prédire le plus de vrais positifs en comparaison aux autres méthodes. Notons que nous n'avons pas considéré dans nos tests les ARNs 16S et 23S car les trois logiciels knotSeeker, pknotsRG, vsfold ne retournaient aucun résultat pour cause de complexité en temps.

Robustesse de Tfold par rapport au type d'ARN considéré

Comme nous l'avons vu plus haut, les résultats obtenus par *Tfold* sont globalement très bons sur l'ensemble des ARN considérés, et homogènes quelque soit l'ARN pour lequel est effectuée la prédiction. La Table 3.5 illustre bien cette particularité de *Tfold*. *Tfold* donne en effet la meilleure sensibilité moyenne (80%), la meilleure sélectivité moyenne (93%) et le meilleur MCC moyen (86 %). Il donne aussi les meilleures valeurs de sensibilité, sélectivité et MCC minimaux. Concernant la sensibilité, la sélectivité et le MCC maximaux, il est parmi les deux meilleurs logiciels. Enfin, *Tfold* a la variabilité la plus faible dans les résultats par rapport aux autres logiciels. Cela confirme l'homogénéité des résultats de prédiction de *Tfold* quel que soit l'ARN considéré.

3.6.3 Conclusion

Tfold est un algorithme de prédiction de structure secondaire basé sur l'approche comparative (intégrant aussi des informations thermodynamiques), mais qui pallie au problème de l'influence des séquences homologues considérées et de la qualité de leur alignement sur le résultat de prédiction, et ce en éliminant, grâce à SSCA, les séquences mal alignées et les séquences trop proches ou trop éloignées en terme de similitude par rapport à la séquence cible.

L'efficacité de Tfold est double : en terme de complexité algorithmique en temps, qui est de $\mathcal{O}(n^2)$, alors que tous les autres algorithmes de la littérature sont en $\mathcal{O}(n^3)$ (voir plus lorsque les pseudonœuds sont recherchés), et en terme de résultats de prédiction, puisque Tfold est le seul à donner, quelque soit l'ARN considéré (en taille, et contenant ou pas de pseudonœuds), des résultats homogènes qui sont toujours supérieurs à 74% en sensibilité et à 87% en sélectivité.

	Sensibilité			Sélectivité			MCC		
	avg	min	max	moy	min	max	moy	min	max
Mfold	0.45	0.17	0.64	0.44	0.17	0.64	0.45	0.17	0.63
*LocARNA**	0.72	0.54	**1**	0.84	0.72	**1**	0.77	0.62	**1**
RNAalifold	0.73	0.55	**1**	0.86	0.66	**1**	0.79	0.64	**1**
*Pfold**	0.76	0.47	**1**	0.89	0.79	**1**	0.82	0.61	**1**
caRNAc	0.49	0.24	0.64	0.88	0.69	**1**	0.62	0.40	0.80
*ILM**	0.64	0.51	0.8	0.72	0.48	**1**	0.68	0.50	0.89
*pknotsRG**	0.53	0.17	**1**	0.53	0.17	**1**	0.53	0.17	**1**
*vsfold**	0.35	0.08	0.90	0.44	0.12	**1**	0.39	0.09	0.95
Tfold	**0.80**	**0.74**	0.95	**0.93**	**0.87**	**1**	**0.86**	**0.81**	0.95

TABLE 3.5 – Les valeurs moyennes, minimales et maximales de sensibilité, de sélectivité et de corrélation (MCC) obtenues par chacun des logiciels sur les différents ARNs considérés : ARNt, ARN 5S, ARN U1, ARN SRP, ARNtm, RNase P, ARN 16S et ARN 23S (les logiciels avec '*' ne donnent pas de résultats pour tous les ARNs).

Tfold a montré une nette performance toujours aussi bien en terme de résultats de prédiction que de rapidité dans le cas de la prédiction de pseudonœuds, en comparaison à plusieurs algorithmes de la littérature spécialisés dans cette problématique.

Enfin, Tfold intègre plusieurs fonctionnalités qui peuvent être très utiles pour les utilisateurs, tels que la possibilité de spécifier certaines hélices éventuellement connues, et la possibilité d'avoir en sortie plusieurs structures alternatives.

Chapitre 4

Prédiction de précurseurs de microARNs

4.1 Introduction

Parce que la détection des miARNs par des techniques expérimentales est difficile et coûteuse et nécessite beaucoup de temps, les méthodes *in silico* représentent la première étape dans l'identification des miARNs. Un nombre important d'outils ont été développés pour prédire d'une part les précurseurs de miARNs et d'autre part les cibles de miARNs, et il existe par ailleurs quelques outils pour la prédiction des miARNs matures.

Nous nous sommes pour notre part intéressés à la prédiction des précurseurs de miARNs, avec comme objectif le développement d'algorithmes efficaces en terme de résultats de prédiction, mais également efficaces en terme de rapidité. En effet, l'un des enjeux actuels en bioinformatique avec les nouvelles technologies de séquençage à hauts débits (les NGS) est le traitement à grande échelle de données génomiques.

4.1.1 Approches existantes et état de l'art

Les méthodes développées pour la prédiction des précurseurs de miARNs peuvent être divisées en trois approches principales : les approches comparatives, les approches par homologie et les approches *ab initio*.

La conservation phylogénétique de certains miARNs dans leur séquence primaire et/ou leur structure secondaire est utilisée dans des approches de génomique comparative. Ces approches considèrent des alignements multiples de séquences où les miARNs conservés sont recherchés. Quelques algorithmes ont été développés, à savoir miRseeker [101], MiRFinder [82], RNAmicro [77], BayesmiARNfind [214], miRRim [187], *MiRScan* [118].

L'augmentation de miARNs connus répertoriés dans miRBase (base de données dédiée aux miARNs [98, 130]), permet une recherche de miARNs homologues à des miARNs déjà connus, en exploitant des informations concernant à la fois la séquence et la structure de ces derniers. On peut par exemple citer dans cette catégorie miRAlign [200] et ERPIN [110].

Les approches à base d'homologie ne peuvent pas détecter les miARNs de familles inconnues et les miARNs sans homologues. En outre, les approches comparatives ne peuvent être utilisées pas sur de nouveaux génomes qui n'ont pas d'espèces voisines séquencées. Les méthodes *ab-initio* sont donc nécessaires pour prédire de nouveaux miARNs dans les génomes. Les approches *ab initio* peuvent être classées en trois catégories :

– La première catégorie prédit les pré-miARNs en considérant d'autres informations, par exemple la position de pré-miARNs voisins déjà connus, comme cela est fait dans *miR-abela* [168] et *MIReNA* [124]. En effet, des pré-miARNs peuvent se présenter sous forme de clusters sur le génome, éventuellement co-transcrits.

– La seconde catégorie recherche les pré-miARNs apparaissant dans les génomes en utilisant des propriétés intrinsèques de la séquence et de la structure des pré-miARNs, comme cela est fait dans CID-miRNA [194], miRPara [208], miRPred [17], miRANK [210] et VMir [63].
– Enfin, la troisième catégorie classifie des pré-miARNs candidats comme vrais ou faux pré-miARNs. Parmi les différentes techniques mises au point pour ce problème de classification, les méthodes d'apprentissage automatique sont les plus prometteuses et les plus utilisées. Plusieurs techniques d'apprentissage automatique ont été appliquées pour classer des pré-miARNs, telles que la programmation génétique (miRPred [17]), les forêts aléatoires (MiPred [87]), les marches aléatoires (miRank [210]), les réseaux bayésiens (BayesmiARNfinder [214]), les modèles de Markov cachés (CSHMM [2], *proMIR* [135]), et les SVM (support vector machine) (triplet-SVM [211], miPred [137] et miRPara [208]).

4.1.2 Problématiques et enjeux

A notre connaissance, il y a très peu d'algorithmes *ab-initio* pour la recherche de structures de pré-miARNs dans des génomes entiers et tous sont spécifiques à un ou plusieurs génomes. Les très rares algorithmes qui font de la recherche de pré-miARNs dans une séquence génomique se limitent considérablement dans la longueur de la séquence considérée, pour cause de complexité en temps. Développer des méthodes rapides capables de traiter des génomes entiers et d'effectuer des prédictions de miARNs à grande échelle est donc un enjeu important de la bioinformatique actuelle. L'autre problème important est la sélectivité de ce type d'algorithmes, qui reste souvent faible, entraînant ainsi un nombre important de faux positifs dans les pré-miARNs prédits. Il est donc important de développer des méthodes permettant de minimiser le nombre de faux positifs, et les méthodes d'apprentissage s'avèrent intéressantes pour cela.

Comme nous l'avons vu plus haut, plusieurs méthodes de classification de pré-miARNs basées sur de l'apprentissage automatique ont été développées. Néanmoins, comme le nombre de pré-miARNs non-déterminés est beaucoup plus élevé que celui des pré-miARNs identifiés, on est confronté à un déséquilibre dans les données d'apprentissage utilisées pour la construction du modèle. Les classifieurs traditionnels basés sur l'apprentissage automatique, tel que les SVM standards, ne sont pas adaptés pour faire face à l'apprentissage sur des données déséquilibrées. Il est donc nécessaire de proposer des méthodes spécifiques pour tenir compte de ce déséquilibre. Quelques méthodes ont récemment été développées pour surmonter le problème de déséquilibre dans les pré-miARNs, tels que *microPred* [11], *MiRenSVM* [41] et plus récemment *HeteroMirPred* [112] et *HuntMi* [66] mais toutes ont des résultats en prédiction non satisfaisants et/ou des temps d'exécution extrêmement longs. Proposer des méthodes de classification de pré-miARNs rapides et efficaces reste donc un problème ouvert.

Une problématique très importante dans la prédiction des pré-miARNs et dans ce type d'algorithmes en général est le choix des caractéristiques à considérer pour l'objet étudié, en l'occurrence ici les pré-miARNs. En effet, selon les caractéristiques considérées, les résultats peuvent différer et être plus ou moins concluants. De plus, dans les méthodes basées sur l'apprentissage automatique, disposer de caractéristiques suffisamment discriminantes entre les vrais et les faux pré-miARNs est l'une des conditions principales pour le bon fonctionnement de la méthode.

4.1.3 Notre contribution

Avec le post-doctorant Sébastien Tempel, nous avons développé une nouvelle méthode *ab-initio* appelée *miRNAFold*, pour la prédiction de structures de pré-miARNs dans les génomes. L'originalité de notre algorithme est qu'il recherche directement les structures en hairpin (épingle à cheveux) correspondant aux pré-miARNs. Les algorithmes existant dans la littérature utilisent des outils de prédiction de structure secondaire tels que *Mfold* [126, 129] ou *RNAFold* [78] puis sélectionnent les hairpins sur lesquelles est appliqué un ensemble de critères. miRNAFold quand à lui vise plus précisément la structure des pré-miARNs en tenant compte de leurs caractéristiques, afin de mieux sélectionner les vrais pré-miARNs et de réduire les temps de

recherche. Nous avons donc défini de nouvelles caractéristiques des structures des pré-miARNs, grâce à une étude statistique des miARNs déjà répertoriés dans miRBase, ces caractéristiques ayant permis de mettre en place une méthode efficace de sélection des structures en hairpin des pré-miARNs. L'idée principale est de rechercher d'abord une longue tige de l'épingle (voir la plus longue), composée d'une longue succession d'appariements, et qui est considérée comme un point d'ancrage permettant de déduire ensuite le reste de la structure en hairpin.

miRNAFold a été testé sur une séquence artificielle et sur plusieurs séquences génomiques réelles. Il réussit à prédire quasi tous les pré-miARNs connus dans les séquences génomiques de différentes espèces. Il possède une sélectivité et une sensibilité de prédiction légèrement meilleures par rapport à ses principaux concurrents, mais il est 60 fois plus rapide. Il met moins de 30 secondes pour analyser une séquence génomique de 1 Mo lorsque le plus rapide des concurrents met 30 minutes. Ce travail a été présenté à la conférence nationale de Bioinformatique JOBIM [186] et publié dans la revue Nucleic Acids Research [185].

Malgré sa grande rapidité, *miRNAFold* reste trop long pour la prédiction des miARNs sur des génomes entiers. La longueur des génomes à étudier varie en effet de quelques dizaines de millions pour les plus petits génomes à quelques milliards pour des organismes évolués. Dans le cadre du projet OpenGPU du pôle de compétitivité System@tic (janvier 2010 - juin 2012) (`http://www.opengpu.net`), nous avons collaboré avec 2 entreprises spécialisées dans le parallélisme et le HPC, à savoir ATEJI et MindsPlanet, à l'optimisation et la parallélisation de *miRNAFold* pour une utilisation sur des machines parallèles, et plus précisément sur des GPU (Unités de Processeurs Graphiques). Une parallélisation partielle de l'algorithme a permis de gagner un facteur d'accélération de 17 (avec une GeForce GTX 580). Ce travail a fait l'objet d'une publication dans un chapitre de livre [180].

Un autre problème de miRNAFold est sa sélectivité faible, le nombre de pré-miARNs prédits sur une longue séquence génomique étant beaucoup trop élevé. Pour améliorer sa sélectivité, nous nous sommes intéressés, avec le post-doctorant Van Du Tran et en collaboration avec Farida Zehraoui de l'équipe AROBAS, à l'utilisation des méthodes d'apprentissage automatique pour déterminer si une séquence donnée correspond à un pré-miARN ou non. Le principe de ces méthodes est la construction de modèles de prédiction en considérant un ensemble d'apprentissage positif et un autre négatif. Différents algorithmes de classification existent dans la littérature, utilisant différentes méthodes d'apprentissage, mais la plupart d'entre eux se basent sur le fait que les deux ensembles d'apprentissage sont de tailles similaires. Or dans les données réelles, l'ensemble des données négatives est plus important que celui des données positives. Nous avons donc développé un algorithme appelé miRBoost permettant de traiter les données déséquilibrées. Il utilise la méthode de boosting associée à une variante faible des SVM. Les SVM représentent une technique de classification largement utilisée dans différents domaines dont la bioinformatique, et ce pour sa grande performance. La méthode ainsi mise en œuvre, combinée à une stratégie de choix et de sélection efficace des caractéristiques des pré-miARNs, a permis d'aboutir à un algorithme favorablement comparable aux méthodes existantes dans la littérature. Notre algorithme donne de très bons résultats en comparaison avec l'état de l'art, sur l'humain et sur un ensemble de données regroupant plusieurs espèces. Il peut être exécuté efficacement dans un délai n'excédant pas quelques minutes, voir quelques secondes (26 secondes pour les données humaines), tandis que presque tous les autres algorithmes existants qui traitent les données déséquilibrées prennent plusieurs heures. Une première version de ce travail a été publié dans la conférence nationale de Bioinformatique JOBIM [192], et une version plus aboutie est en cours de révision pour une soumission au journal international RNA.

4.2 *miRNAFold* : Recherche *ab-initio* de précurseurs de microARNs dans les génomes

Nous avons développé une nouvelle méthode *ab initio* appelée *miRNAFold* pour la prédiction de la structure en épingle à cheveux de pré-miARNs dans les génomes. Notre objectif était de concevoir un algorithme qui soit capable de trouver efficacement les pré-miARNs dans des génomes entiers dans un délai raisonnable. Pour ce faire, nous recherchons directement les structures en hairpin des pré-miARNs, en appliquant des critères de sélection à différentes étapes de la recherche, et ce afin d'éliminer très vite d'éventuels candidats faux positifs. Une première étape de notre travail a donc été d'étudier les structures des pré-miARNs connus afin de mettre en évidence des caractéristiques communes.

4.2.1 Caractéristiques des précurseurs de miARNs

Notre premier objectif a été de trouver des caractéristiques relatives aux hairpins de pré-miARNs, dont un exemple est donné en Figure 4.1.

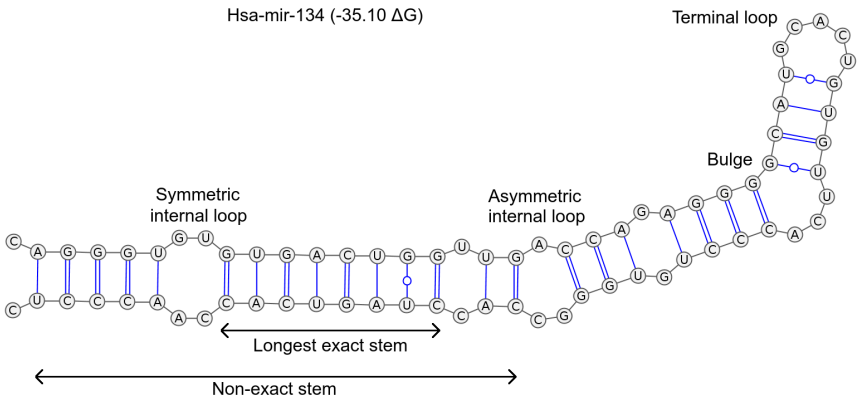

FIGURE 4.1 – Exemple de structure en épingle à cheveux (hairpin).

Pour cela, nous avons étudié les 16 772 pré-miARNs contenus dans la base de données de miRBase (version 17, April 2011). Nous avons alors observé plusieurs caractéristiques (voir Figure 4.2) :

Les structures des pré-miARNs contiennent de longues hélices : Nous avons observé que les pré-miARNs sont presque toujours composés d'au moins une longue tige exacte. Et comme nous pouvons le voir dans la Figure 4.2 (A), la plus longue tige exacte de la structure en épingle à cheveux des pré-miARNs de miRBase est souvent entre 5 et 10 nt.

Les structures des pré-miARNs sont symétriques : Nous avons également observé que la plupart des pré-miARNs ont soit très peu de renflements, soit des renflements d'un côté compensés avec des renflements de l'autre côté (c'est à dire qu'il y a un nombre similaire de nucléotides des deux côtés de l'épingle à cheveux de la boucle terminale aux extrémités). Figure 4.2 (B) montre que le nombre d'épingles à cheveux diminue lorsque l'écart entre les deux brins de l'hairpin augmente. 90% des pré-miARNs de miRBase ont moins de 3 nucléotides de plus sur un côté. En d'autres termes, les pré-miARNs ne forment pas une épingle à cheveux "courbée" mais plutôt "droite".

Les structures des pré-miARNs peuvent être approximées par une hélice non-exacte : Nous avons constaté que dans presque tous les pré-miARNs de miRBase il y a une tige non exacte. Une tige non-exacte est composée d'une succession de tiges exactes séparées par des boucles symétriques et telle que la taille de chaque boucle symétrique est inférieure à la longueur exacte de la tige qui l'entoure (la taille d'une boucle symétrique est le nombre de nucléotides non appariés sur un côté de la boucle). Plus de 75% de pré-miARN

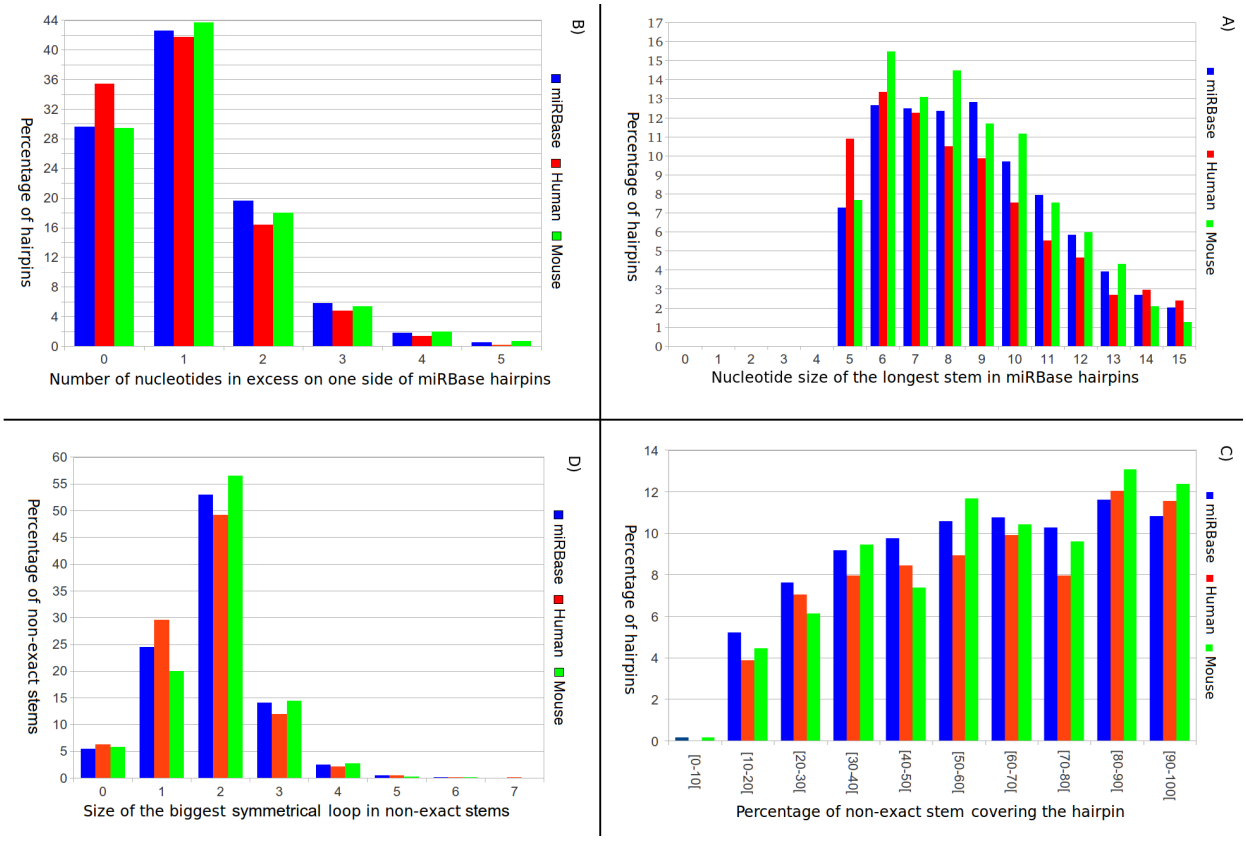

FIGURE 4.2 – Pourcentage des épingles à cheveux de pré-miARNs dans le génome humain, le génome de la souris et dans tout miRBase : (A) en fonction de la longueur de leur plus longue hélice ; (B) ayant un intervalle d'une taille donnée, c'est à dire ayant un excès de nucleotides d'un côté de l'épingle à cheveux (un écart de zéro correspond à un même nombre de nucléotides des deux côtés) ; (C) en fonction du pourcentage de nucléotides couverts par une tige non exacte ; (D) en fonction de la taille de leur plus grande boucle symétrique.

dans miRBase ont une tige non-exacte qui représente au moins 40% de leur longueur (Figure 4.2 (C)). Ce pourcentage correspond au rapport entre la taille de la tige non-exacte et la taille de l'épingle à cheveux (sans la boucle terminale).

Presque tous les pré-miARNs de miRBase ont de courtes boucles symétriques. 91,5% des pré-miARNs ont des boucles symétriques dont la longueur varie de 1 à 3 nucléotides. Seulement 0,15% des pré-miARNs de miRBase (24 sur 16 772) ont une boucle symétrique de longueur supérieure ou égale à 6 (Figure 4.2 (D)).

Autres caractéristiques des pré-miRNAs : En étudiant les pré-miARNs de miRBase, nous avons observé plusieurs autres caractéristiques, qui peuvent être divisées en deux catégories : les caractéristiques globales qui sont présentes dans toutes les espèces de miRBase et les caractéristiques spécifiques à chaque espèce. Pour les caractéristiques globales, nous avons observé par exemple que les plus longues tiges présentent un pourcentage de paires de bases GU toujours inférieur à 33,33%. Nous avons également observé que la taille moyenne des tiges exactes qui constituent les tiges non-exactes est supérieure à 3.

Pour les caractéristiques spécifiques aux espèces, nous avons utilisé certaines caractéristiques habituelles comme la taille de l'épingle à cheveux, l'énergie libre minimale (MFE) et le rapport entre les nucléotides A, C, G et U. Le paramètre MFE est calculé de la même façon que dans Mfold [190]. Nous avons également calculé certaines caractéristiques à partir de Helvik *et al.* [107] et van der Burgt *et al.* [70] comme le MFE ajusté (c'est à dire le rapport entre le MFE et la longueur), le rapport entre les appariements GU et GC et le rapport de G sur C.

4.2.2 Notre approche et algorithme pour l'identification des miARNs

Nous considérons une fenêtre glissante de taille L, suffisamment longue pour contenir un pré-miARN, dans laquelle nous essayons de détecter des structures en épingle à cheveux de pré-miARNs. Dans un premier temps, nous cherchons les longues hélices exactes qui vérifient certains critères. Celles-ci sont alors considérées comme des "ancres" de possibles épingles à cheveux. Dans une deuxième étape, nous étendons les hélices (ou tiges) sélectionnées afin d'obtenir les plus longues hélices non-exactes satisfaisant certaines contraintes. Chaque tige non-exacte sélectionnée peut être considérée comme une bonne approximation d'une épingle à cheveux de pré-miARN, et fournit ainsi la position d'une possible épingle à cheveux qui est ensuite modélisée. Ainsi, notre approche consiste en trois étapes principales appliquées sur chaque sous-séquence de la fenêtre :

1. Recherche des plus longues tiges exactes.
2. Extension des tiges sélectionnées puis sélection des plus longues tiges non exactes.
3. Prédiction de structures en épingles à cheveux correspondant aux tiges non exactes sélectionnées.

Etant donnée une séquence génomique, une matrice triangulaire M d'appariements de bases est construite pour chaque sous-séquence délimitée par la fenêtre coulissante, où la sous-séquence est comparée à son inverse. L'algorithme effectue ensuite les trois étapes principales données ci-dessus et illustrées dans la Figure 4.3. A chacune de ces trois étapes, un certain pourcentage de critères de sélection est appliqué : aux tiges exactes, puis aux tiges non-exactes, et enfin aux hairpins.

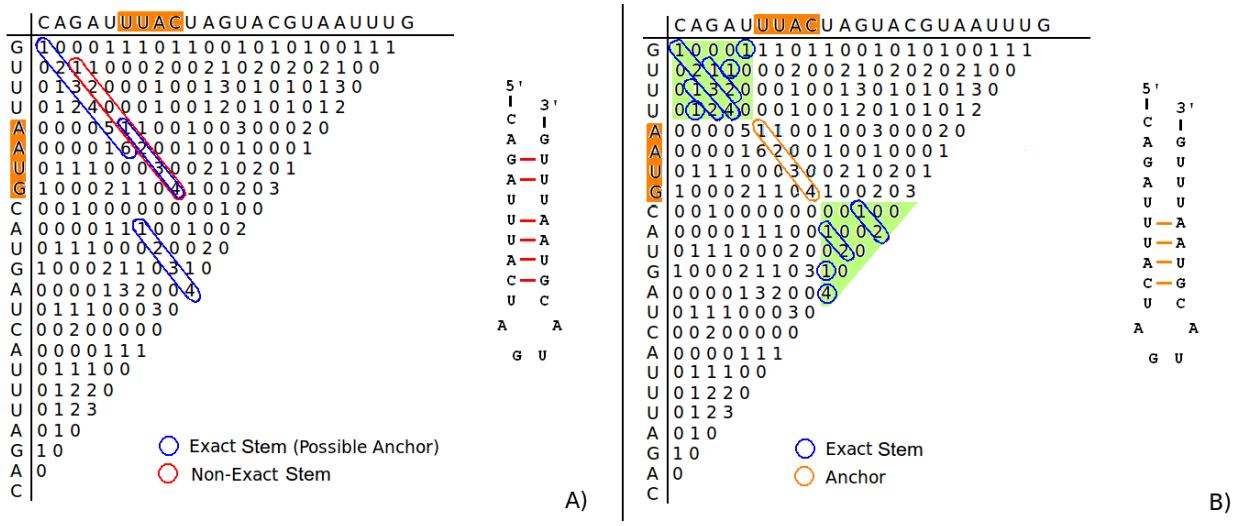

FIGURE 4.3 – (A) Exemple d'une matrice pour la détermination d'hélices exactes et non exactes dans une sous-séquence génomique donnée. Les trois plus longues tiges sont sélectionnées (entourées d'un cercle bleu). L'une des trois tiges a été étendue à une tige non-exacte (entourée d'un cercle rouge). (B) Recherche des épingles à cheveux. L'ancre (entouré par un cercle orange) de la tige non-exacte représentée en (A) est positionnée dans la matrice, puis est étendue à gauche et à droite (zones vertes) sur différentes diagonales, de façon à permettre des renflements et boucles internes.

4.2.3 Résultats

miRNAFold a été testé sur une séquence artificielle et sur plusieurs séquences génomiques réelles et a été comparé à différentes méthodes de même catégorie existantes dans la littérature, à savoir des méthodes permettant la recherche *ab initio* de pré-miARNs dans de longues séquences génomiques. Nous pouvons citer cinq méthodes dans cette catégorie : CID-miRNA [194], miRPara [208], miRPred [17], miRANK

[210] et VMir [63]. Malheureusement, nous ne pouvions pas accéder au code source, binaire ou serveur web de miRPred et miRANK. Nous avons donc considéré CID-miRNA, miRPara et VMir pour nos tests.

Nous avons par ailleurs considéré triplet-SVM [211], un logiciel initialement prévu pour classifier des séquences en pré-miARNs ou non pré-miARNs, que nous avons adapté pour une recherche de pré-miARNs dans des séquences génomiques.

Un pré-miARN prédit ne correspond pas toujours exactement au pré-miARN fonctionnel. Par conséquent, nous considérons qu'un pré-miARN connu est correctement prédit si la position retournée est correcte. La position d'une épingle à cheveux est considérée comme le centre et nous supposons qu'un pré-miARN prédit correspond à un pré-miARN connu si la distance entre le connu et le centre du pré-miARN prédit est inférieur à 10% de la taille de l'épingle à cheveux.

Afin d'évaluer et de comparer les différents programmes testés, nous avons utilisé les mesures de sensibilité et de sélectivité. La sensibilité mesure la capacité du logiciel à trouver les pré-miARNs connus. La sélectivité représente la probabilité qu'une épingle à cheveux prédite correspond à un pré-miARN. La sensibilité et la sélectivité sont donnés respectivement par les équations $\frac{TP}{TP+FN}$ et $\frac{TP}{TP+FP}$, où TP (vrais positifs) est le nombre de pré-miARNs connus prédits, FN (faux négatifs) est le nombre de pré-miARNs connus non prédits, et FP (faux positifs) est le nombre de pré-miARNs prédits correspondant à des non pré-miARNs.

Résultats sur une séquence artificielle

Un paramètre important de miRNAFold est le pourcentage minimal de critères qui doivent être vérifiés à chaque étape de l'algorithme. Pour déterminer ce paramètre, nous avons testé miRNAFold sur une séquence artificielle.

La séquence artificielle est créée par la concaténation des ARNm humains et l'insertion de 100 pré-miARNs humains. Les séquences d'ARNm proviennent du génome humain (build 37.2) du site NCBI (`www.ncbi.nlm.nih.gov`) et les pré-miARNs proviennent de la base de données miRBase (release 17) [130]. La longueur des pré-miARNs sont de 63 à 110 nt et la position de début des pré-miARNs commence chaque 300 nt, le premier pré-miARN commençant à la position 300. La longueur totale de la séquence artificielle est de 30 500 nucléotides.

Les résultats de sensibilité et de sélectivité obtenus par miRNAFold, CID-miRNA, miRPara, triplet-SVM et Vmir, ainsi que leurs temps d'exécution sont donnés dans la Table 4.1.

	Sensibilité	Sélectivité	Temps d'exécution
CID-miRNA	0.97	0.12	90m49s
miRPara	0.97	0.10	5m24s
triplet-SVM	1.00	0.05	5m
VMir	0.28	0.01	2m32s
miRNAFold_{50}	**0.98**	0.19	0.88s
miRNAFold_{60}	**0.98**	0.19	0.88s
miRNAFold_{70}	0.97	0.19	0.84s
miRNAFold_{80}	0.96	0.23	0.76s
miRNAFold_{90}	0.65	**0.52**	**0.68s**

TABLE 4.1 – Les résultats obtenus par miRNAFold, CID-miRNA, miRPara, Triplet-SVM et Vmir sur la séquence artificielle. miRNAFold a été exécuté avec différentes valeurs pour le paramètre de pourcentage minimal de critères vérifiés : 50%, 60%, 70%, 80% and 90%.

Comme on peut le voir, miRNAFold est largement plus rapide que les autres méthodes testées, et il est toujours plus sélectif, quel que soit le critère de pourcentage considéré. Comme prévu, plus la valeur du

paramètre de pourcentage dans miRNAFold est faible, plus la sensibilité est élevée. Plus la valeur de ce pourcentage est élevée, plus la sélectivité est élevée. Quand on augmente le pourcentage de critères de 50% à 80%, miRNAFold manque seulement 2 pré-miARNs, mais supprime plus de 109 faux pré-miARNs. Ces seuils permettent à l'utilisateur, et de façon simple, de privilégier soit la prédiction d'un maximum de pré-miARNs, soit la prédiction d'un minimum de faux positifs. Dans la suite, nous avons fixé à 70% la valeur par défaut de ce paramètre, car il représente le pourcentage donnant un bon compromis entre la sensibilité et la sélectivité.

Résultats sur des séquences génomiques réelles

Nous avons testé miRNAFold ainsi que CID-miRNA, miRPara et VMir sur les quatre séquences génomiques : humain, souris, poisson zèbre et Sea squirt. Nous avons choisi ces génomes car ils présentent chacun un grand cluster de miARNs connus :

– Le chromosome humain 19 (brin '+') a un cluster de 50 pré-miARNs, le premier pré-miARN commençant à la position 54.169.933 et le dernier se terminant en position 54.485.651.
– Le chromosome 2 de la souris (brin '+') a un cluster de 71 pré-miARNs, le premier commençant à la position 10.388.290 et le dernier se terminant en position 10.439.906.
– Le chromosome 4 du poisson zèbre (brin '-') a un cluster de 50 pré-miARNs, le premier commençant à la position 34.353.975 et le dernier se terminant en position 34.481.435.
– Le chromosome 7q du Sea quirt (brin '-') a un cluster de 46 pré-miARNs, le premier commençant à la position 5.400.066 et le dernier se terminant en position 6.168.570.

Pour chacun de ces génomes, nous avons extrait du site de NCBI la sous-séquence qui comprend le cluster. Table 4.2 montre les résultats de sensibilité et de sélectivité obtenus par miRNAFold, CID-miRNA, miRPara, triplet-SVM et VMir sur chacune de ces séquences et Table 4.3 donne les différents temps d'exécution [1].

	Sensibilité				Sélectivité			
	Humain	Souris	Zebrafish	Sea squirt	Humain	Souris	Zebrafish	Sea squirt
CID-miRNA	38	29. 58	19.30	28.26	0.69	0.82	0.75	**10.88**
miRPara	98	**98.59**	47.37	58.7	**0,93**	5.34	1.4	5.86
VMir	**100**	88.73	84.21	**100**	0.56	2.93	1.35	5.29
triplet-SVM	**100**	38	6.5	20	0.45	1.36	0.07	0.04
miRNAFold$_{70}$	**100**	**98.59**	**94.74**	91.30	0.89	**7.71**	**2.60**	7.98

TABLE 4.2 – Résultats de sensibilité et de sélectivité (les valeurs sont multipliées par 100) obtenus par CID-miRNA, miRPara, VMir et miRNAFold sur les séquences génomiques de l'Humain, de la Souris, du Zebrafish et du Sea squirt.

Comme on peut le voir sur la Table 4.2, miRNAFold a une meilleure sensibilité et sélectivité que les trois autres algorithmes sur les séquences de souris et de poisson zèbre. Dans la séquence génomique humaine, miRPara a une meilleure sélectivité que miRNAFold, mais il a une plus faible sensibilité. Dans la séquence génomique de Sea squirt, miRNAFold est le seul algorithme qui donne une sensibilité toujours supérieure à 90 % pour toutes les séquences testées. Contrairement aux autres programmes, il donne des résultats de sensibilité homogènes et stables quelle que soit la séquence génomique considérée.

miRNAFold est l'algorithme le plus rapide. Les temps d'exécution de miRNAFold sont de l'ordre de quelques secondes (25 s) pour une séquence de taille de 1 million de nucléotides, lorsque VMir, le deuxième algorithme le plus rapide, a un temps d'exécution de de plusieurs minutes ($\sim$ 30 mn).

1. Les expériences ont été effectuées sur une machine Linux équipée de processeurs Intel Core Duo 2 T6600 de 2,2 GHz et 4 Go de RAM.

	Humain	Souris	Zebrafish	Sea squirt	Moyenne
CID-miRNA	54h58m	54h48m	54h40m	55h29m	55h08m
miRPara	20h12m	19h47m	19h40m	19h25m	19h46m
triplet-SVM	50m	15m	30 m	1h40	48m
VMir	30m	30m	30m	30m	30m
miRNAFold$_{70}$	**0m25s**	**0m22s**	**0m29s**	**0m24s**	**0m25s**

TABLE 4.3 – Temps d'exécution des algorithmes CID-miRNA, miRPara, triplet-SVM, VMir et miRNAFold pour la prédiction de pré-miARNs dans les séquences génomiques de 1 million de nucléotides dans chacune des quatre espèces Humain, Souris, Zebrafish et Sea squirt.

4.2.4 Passage à l'échelle : version GPU de l'algorithme *miRNAFold*

Les nouvelles générations de séquençage créent une quantité de données énormes, dont le traitement avec les méthodes traditionnelles sur CPU deviennent excessivement coûteuses en temps de calcul. La parallélisation des algorithmes de bioinformatique pour le traitement de ces données devient donc nécessaire. Néanmoins, la plupart des laboratoires de recherche (publics ou privées) n'ont pas les moyens d'acheter des gros clusters de calculs CPU. C'est pourquoi l'utilisation d'ordinateurs contenant des Unités de Processeurs Graphiques (GPU) sont une alternative réaliste au prix des clusters de PC, pour une grande puissance de calculs avec un coût matériel raisonnable.

Malgré sa rapidité, miRNAFold reste trop long pour une recherche dans des génomes entiers, notamment les grands génomes. Un grand challenge est donc de pouvoir réduire encore considérablement le temps d'exécution de ce type d'algorithmes. Dans le cadre du projet OpenGPU (FUI 8, pôle de compétitivité System@tic), et en collaboration avec Eric Mahé de MindsPlanet, nous avons développé une version GPU de miRNAFold pour une meilleure performance en temps d'exécution.

Parallélisation de miRNAFold

L'algorithme *miRNAFold* divise la séquence d'entrée en petites séquences et calcule une matrice d'appariement de bases pour chaque sous-séquence, où les pré-miARNs candidats sont ensuite recherchés (voir Section 4.2.2). Etant donné que les pré-miARNs candidats sont indépendants les unes des autres, leurs opérations de prédiction sont indépendantes, et donc les matrices d'appariement sont également indépendantes les unes des autres. Par conséquent, les étapes préliminaires de *miRNAFold*, c'est à dire diviser en sous-séquences et calculer les diagonales de la matrice d'appariement, peuvent être facilement parallélisées (voir Figure 4.4).

Etant donnée une séquence, *miRNAFold* calcule le nombre de sous-séquences nécessaires à l'identification de tous les pré-miARNs possibles de la séquence. Ce nombre est défini par la taille de la séquence divisée par la taille du décalage de chaque fenêtre glissante, par défaut égale à 10 nucléotides. *miRNAFold* a été développé pour la recherche de pré-miARNs dans des chromosomes entiers. Avec une taille de chromosome eucaryote moyenne d'une centaine de millions de nucléotides, *miRNAFold* devrait créer 10 millions de sous-séquences. D'autre part, la taille de la sous-séquence correspond à la taille maximale des pré-miARNs recherchés. Cette taille maximale varie de 50 à 300 nucléotides [62]. Par ailleurs, le nombre de diagonales dans la matrice est deux fois plus grand que la taille de la sous-séquence correspondant (Figure 4.4). Le nombre de diagonales et le nombre de sous-séquences correspondent à deux échelles différentes : de quelques milliers à quelques millions pour le nombre de sous-séquences ; de 100 à 1 000 pour le nombre de diagonales de la matrice. Il y a aussi deux échelles sur les cartes GPU : le nombre d'unités d'exécution (threads) disponibles et le nombre de blocs disponibles. Le nombre maximal de threads disponibles est égal à 1 024, et Le nombre maximal de blocs disponibles sur les cartes GPU est autour de 4 000 milliards. Pour paralléliser *miRNAFold* pour une utilisation sur des GPU, nous avons considéré les deux niveaux suivants

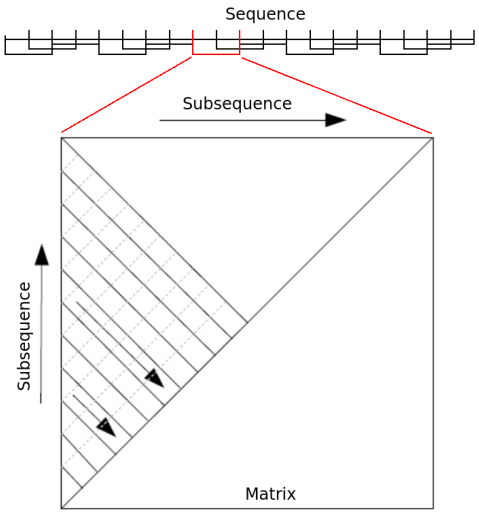

FIGURE 4.4 – Vue d'ensemble des différentes étapes indépendantes de l'algorithme *miRNAFold*. Tout d'abord, *miRNAFold* divise la séquence en sous-séquences. Ensuite, pour chaque séquence, *miRNAFold* construit une matrice d'appariement de bases, où chaque diagonale est traitée indépendamment.

de l'algorithme : les séquences et les diagonales. Ainsi, un thread correspond à la diagonale de la matrice d'appariement et le bloc correspond à la sous-séquence (qui correspond à l'ensemble des diagonales).

miRNAFold est implémenté dans le langage C. Pour faciliter la conversion du code CPU de *miRNAFold* dans un code GPU, nous avons choisi le langage CUDA [34], ce qui permet d'utiliser des codes C à l'intérieur d'instructions CUDA. Une autre raison de ce choix est que nous nous sommes fournis avec une carte Nvidia Quadro 5000 et une Nvidia NVS 4200M, et ce langue est adapté pour la carte GPU Nvidia.

Résultats

Après avoir vérifié que la version GPU de *miRNAFold* a donné la même sortie que la version CPU, nous avons comparé le temps d'exécution des deux versions. Nous avons testé nos algorithmes sur deux ordinateurs avec deux cartes GPU comme suit :

– Ordinateur 1; CPU Intel Xeon W3565, 3.2 GHz
– Ordinateur 1; GPU Nvidia Quadro 5000, 352 CUDA Cores 6 MHz
– Ordinateur 2; CPU Intel i7 2760QM, 2.4 GHz
– Ordinateur 2; GPU Nvidia NVS 4200M, 48 CUDA Cores 810 MHz

Les temps d'exécution de miRNAFold sur les deux ordinateurs sont présentés dans la Figure 4.5. L'augmentation de la vitesse entre l'exécution sur GPU et l'exécution sur CPU est constante : $\times 1.34$ pour la carte NVS 4 200M GPU et $\times 9.12$ pour la carte Quadro 5 000 GPU. Pour une séquence de taille inférieure à 50 000 nucléotides, la différence de temps d'exécution entre les versions CPU et GPU n'est pas significatif, car le transfert de données depuis le disque dur vers la mémoire GPU et vice versa prend un certain temps. Pour une séquence de plus de 50 000 nucléotides, la différence de temps d'exécution devient considérable. Par exemple, le temps d'exécution d'une séquence de 5 millions de nucléotides est de 25 secondes sur le processeur Intel Xeon W3565 et 3 secondes sur GPU Nvidia Quadro 5000. Ces résultats montrent que la version GPU est beaucoup plus rapide que la version CPU.

Comme nous pouvons le voir sur la Figure 4.5, la version GPU de miRNAFold est limitée à une séquence de 2 millions de nucléotides sur GPU Nvidia NVS 4200M. Elle est également limitée à une séquence de 120 millions de nucléotides sur GPU Nvidia Quadro 5000. Ces limitations sont dues à la limite des mémoires

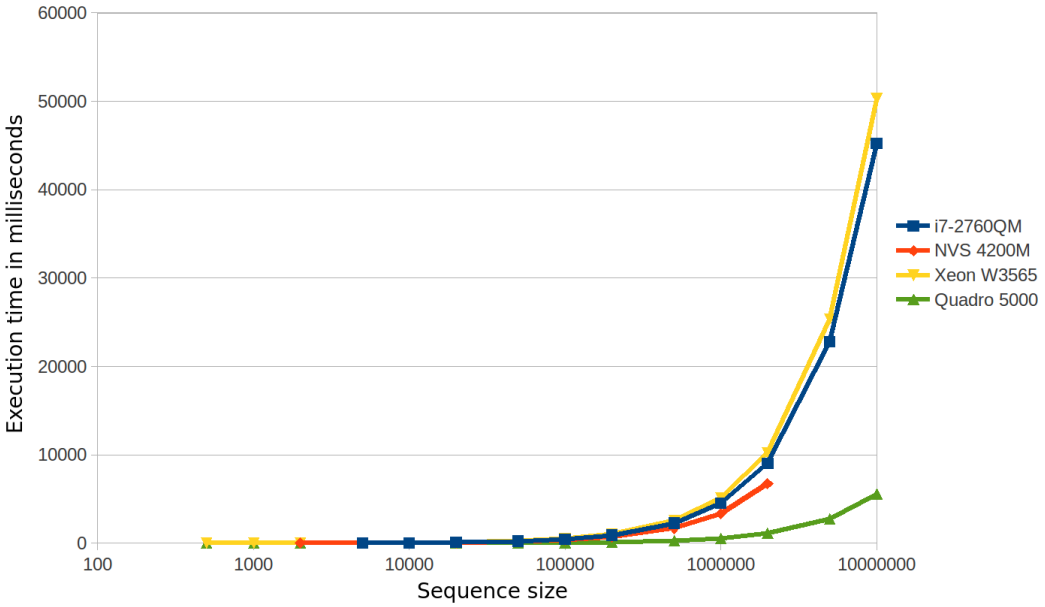

FIGURE 4.5 – Comparaison entre le temps d'exécution de *miRNAFold* sur CPU et sur GPU. Le graphique utilise des échelles logarithmiques. Les différentes lignes correspondent aux différents temps d'exécution CPU et GPU.

disponibles sur les cartes GPU. L'utilisation de plusieurs cartes GPU devrait donc permettre de surmonter ce problème.

Nous avons par ailleurs effectué des tests avec la carte Nvidia GeForce GTX 580. Cette carte GPU était l'une des meilleurs cartes GPU de Nvidia au moment où nous avions effectué ces tests. La carte GPU a été branchée à l'ordinateur 2. L'augmentation de la vitesse des temps d'exécution des versions GPU et CPU est ici aussi constante : $\times 16,75$ pour la carte GPU GeForce GTX 580.

4.2.5 Conclusion

Nous avons développé une méthode *ab initio* originale appelée miRNAFold qui permet une recherche rapide des précurseurs de miARNs dans les génomes. En utilisant une fenêtre glissante, notre méthode recherche d'abord la position des pré-miARNs en approximant leur structure avant de déduire la structure finale. L'intérêt de cette première étape est de réduire considérablement le temps d'exécution. L'algorithme a une complexité en temps de $\mathcal{O}(L^2 \times N)$, où L est la longueur de la fenêtre, et N la taille de la séquence.

miRNAFold a été testé sur une séquence artificielle et sur plusieurs séquences génomiques réelles, et a été comparé à CID-miRNA, miRPara et VMir. Un avantage important de miRNAFold par rapport à ses concurrents est sa rapidité. Il est 60 fois plus rapide que l'algorithme le plus rapide testé, à savoir Vmir. miRNAFold prend moins de 30 secondes pour une séquence de longueur de 1 million, lorsque VMir prend plus de 30 minutes, miRPara prend environ 20 heures et CID-miRNA plus de 55 heures. En outre, la parallélisation partielle de miRNAFold pour une utilisation sur des machines GPU a permis de rendre miRNAFold 17 fois plus rapide, montrant l'apport que peut avoir les cartes GPU pour de telles applications. Par contre, les tests que nous avons effectués révèlent l'importance de la carte GPU pour le temps d'exécution : l'augmentation de la vitesse entre la carte GPU Nvidia GTX 580 et la carte GPU Nvidia Quadro 5000 est de 2 fois. Ainsi, l'utilisation des nouvelles générations de carte GPU combinée à une parallélisation optimisée et complète de miRNAFold permettrait sans aucun doute d'augmenter encore la rapidité de miRNAFold.

miRNAFold réussit à prédire quasi tous les pré-miARNs connus dans les séquences génomiques de diffé-

rentes espèces. Sa sensibilité est presque toujours meilleure que ses concurrents. C'est le seul algorithme donnant une sensibilité toujours supérieure à 90%. Contrairement aux autres algorithmes testés, sa sensibilité est homogène et stable quelque soit la séquence génomique considérée. Toutefois, la sélectivité de miRNAFold n'est pas satisfaisante. Décroître de manière significative le nombre de faux positifs est l'un des challenges sur lequel nous travaillons. L'une des pistes est l'utilisation de méthodes d'apprentissage automatique permettant une meilleure sélection des pré-miARNs vérifiant des critères similaires à des pré-miARNs déjà connus et validés. Ce travail a fait l'objet du développement d'un algorithme appelé miRBoost, que nous présentons en Section 4.3.

4.3 miRBoost : Classification des vrais et faux précurseurs de microARNs

Afin d'améliorer la sélectivité de miRNAFold, nous avons développé un algorithme, basé sur de l'apprentissage automatique, pour classer des pré-miARNs candidats en pré-miARNs ou non-miARNs.

4.3.1 Problème des données déséquilibrées

Comme le nombre de séquences pré-miARNs est beaucoup plus élevé que celui des séquences de pré-miARNs identifiés, nous sommes confrontés à un déséquilibre dans les données d'apprentissage. Les classifieurs traditionnels basés sur l'apprentissage automatique, tel que les SVM standards, ne sont pas adaptés pour faire face à l'apprentissage sur des données déséquilibrées, car ils ont tendance à classer la plupart des échantillons donnés dans la classe des données les plus abondantes [206]. Par conséquent, la majorité des séquences candidates risquent d'être prédites comme étant des non-miARNs.

Plusieurs approches ont été proposées dans la littérature pour traiter les ensembles de données déséquilibrés dans les SVM. Il existe principalement deux classes de méthodes : les méthodes internes qui introduisent des modifications dans la formulation des SVM pour tenir compte du déséquilibre dans les données [133, 115, 117] et les méthodes externes basées sur des techniques de sur-échantillonage (over-sampling) et de sous-échantillonage (under-sampling) [25, 86, 5]. Certaines méthodes d'ensemble, sans objectifs de traitement de ce problème, ont montré une bonne performance dans le traitement des données déséquilibrées, comme le bagging [18] et le boosting [166], et le boosting a empiriquement été prouvé être plus efficace lorsque les données ne contiennent pas beaucoup de bruit [12, 145].

Quelques méthodes ont récemment été développées pour surmonter le problème de déséquilibre dans les pré-miARNs. *microPred* [11] utilise un classifieur SVM qui est capable de faire face au problème de déséquilibre via des techniques internes et externes. Les méthodes externes ont également été exploitées dans *mirExplorer* [65] qui a aussi utilisé les techniques de boosting pour améliorer les classifieurs faibles. *HeteroMirPred* [112] utilise une combinaison de différents méthodes de classification : les SVM, les k-plus proches voisins et les forêts aléatoires, renforcée par une méthode externe pour la prise en compte du déséquilibre. *MiRenSVM* [41] a utilisé un classifieur SVM combiné à une technique de bagging pour traiter les données déséquilibrées.

4.3.2 Méthode de boosting

Les méthodes de Boosting constituent une famille d'algorithmes d'apprentissage automatique qui visent à rendre un algorithme d'apprentissage dit "faible" (weak classifier) plus performant. Ces méthodes permettent de combiner les résultats de classifieurs faibles sur un ensemble de données considérées difficiles à apprendre, afin de construire un seul classifieur "fort" (strong classifier). Dans le cas où on dispose de deux classes, un classifieur faible est un algorithme qui est capable de reconnaître les classes au moins aussi bien qu'une estimation aléatoire (c'est-à-dire qu'il ne se trompe pas plus d'une fois sur deux en moyenne, si la distribution des classes est équilibrée) [156, 116].

AdaBoost (ou Adaptive Stimuler) est la méthode de boosting la plus populaire proposée dans [52], elle consiste à transformer, d'une manière efficace, un classifieur faible en un classifieur fort en réduisant les taux d'erreur. L'algorithme Adaboost appelle à chaque itération un algorithme d'apprentissage qui entraîne les instances à classifier. Par la suite, Adaboost définit une nouvelle distribution pour les instances d'apprentissage en fonction des résultats de l'algorithme à l'itération précédente tout en augmentant le poids des instances qui ne sont pas correctement classées. A la fin, il combine les résultats des différents classifieurs faibles par un vote pondéré pour en déduire un classifieur fort.

Le choix du classifieur faible dans la méthode AdaBoost est crucial pour son bon fonctionnement. Le SVM, étant un classifieur relativement fort, ne semble pas être adapté au principe de boosting et peut conduire à une dégradation des performances [203]. Cependant, l'utilisation d'une variante faible de SVM permet de donner de très bon résultats, comme indiqué dans [156, 116, 189, 199]. Le boosting sur les classifieurs SVM "faible" peut être aussi efficace que les SVM et montrer une meilleure performance de généralisation par rapport aux SVM.

Différentes méthodes pour construire des classifieurs faibles à partir des SVM ont été proposées dans la littérature [116, 189, 199]. Malgré les bon résultats obtenus lors des évaluations expérimentales, ces méthodes ne garantissent pas que les classifieurs utilisés dans adaBoost soient faibles. Nous nous somme inspirés dans notre travail de l'approche proposée dans Rangel *et al.* [156] pour rendre les SVM faibles, en utilisant à chaque fois des sous-ensembles de données de telle manière que les erreurs d'apprentissage des classifieurs SVM ainsi construits soient contrôlées.

4.3.3 Notre approche

Nous avons implémenté une variante de LIBSVM [24] qui prend en compte les instances pondérées. Ceci permet de pénaliser le déséquilibre entre les échantillons d'apprentissage via leurs différents poids. Le boosting avec ces classifieurs SVM faibles peut améliorer le temps de calcul de l'algorithme d'apprentissage, étant donné que l'apprentissage est réalisé sur un sous ensemble de données pour obtenir des variantes faibles des SVM et nécessite un plus petit nombre de vecteurs de support.

Soit $\mathcal{S} = \{(x_i, y_i)\}_{i \in I}$ un ensemble d'apprentissage étiqueté, avec $x \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$ et $\mathcal{W} = \{w_1, w_2, ..., w_N\}$ une distribution de poids sur $\mathcal{S}$, $\sum_{i \in I} w_i = 1$, avec $I = \{1, 2, ..., N\}$. Pour construire un classifier SVM faible sur $\mathcal{S}$, nous utilisons un sous-ensemble d'apprentissage $\mathcal{J} = \{(x_i, y_i)\}_{i \in J}$, tel que :

$$\sum_{i \in I \setminus J} w_i = \mu \leq \mu_0,$$

où J a une cardinalité minimale, $J \subset I$ et $0 < \mu \leq \mu_0 < 1$.

Notre algorithme pour la construction de classifieurs SVM faibles est décrit comme suit :

Algorithme WeakSVM
- **Entrée :** ensemble d'échantillons étiquetés $\mathcal{S} = \{(x_i, y_i)\}_{i \in I}$; distribution de poids $\mathcal{W} = \{w_i\}_{i \in I}$, $I = \{1, \ldots, N\}$; paramètre μ_0;
- Sélectionner $J \in I$ tel que $\sum_{i \in J} w_i \geq 1 - \mu_0$ et $\mathcal{J}$ a une cardinalité minimale.
- Apprendre un composant classifieur SVM h_t sur $\mathcal{J} = \{(x_i, y_i)\}_{i \in J}$ avec une distribution de poids $\mathcal{W}_J = \left\{ \frac{w_i}{\sum_{j \in J} w_j} \right\}_{i \in J}$.
- **Sortie :** h_t

4.3.4 Sélection des caractéristiques utilisées

Une séquence donnée est identifiée comme étant soit un pré-miARN soit un non pré-miARN sur la base de ses caractéristiques. Il est donc important de choisir un ensemble approprié de caractéristiques de pré-miARNs pour la classification. Dans ce travail, nous utilisons comme caractéristiques les propriétés intrinsèques de la séquence et de la structure des séquences données. Nous avons introduit 62 nouvelles caractériqtiques, parmi lesquelles 26 sont utilisées dans *miRNAFold*. Ces caractéristiques décrivent les propriétés intrinsèques de la structure en épingle à cheveux des pré-miARNs : la taille, l'énergie, la composition (en moyenne et en total) en nucléotidique des tiges exactes et non exactes, la taille et le nombre de renflements et de boucles internes, l'asymétrie de l'hairpin. Nous avons également extrait 125 caractéristiques à partir de la littérature, qui sont utilisées dans plusieurs algorithmes de prédiction de pré-miARNs, dont *microPred* [11], *MiPred* [87], *miR-abela* [168], *miRank* [210], and *triplet-SVM* [211].

Ainsi, nous avons étudié un total de 187 caractéristiques, qui donnent des informations sur la structure et la séquence des pré-miARNs. Elles peuvent être regroupées en trois groupes : celles associées au tiges exactes, celles associées aux tiges non-exactes et celles associées aux hairpins. Pour chaque ensemble de données, et dans le but de choisir parmi ces 187 caractéristiques celles qui sont cohérentes et non-redondantes, nous avons exploité les techniques de sélection de caractéristiques proposées par le workbench WEKA [68]. Nous avons ainsi choisi les caractéristiques découvertes par au moins deux de ces techniques.

Ce processus de sélection de caractéristiques est réalisé de deux façons. Tout d'abord, les ensembles des caractéristiques sont déterminés avec la validation croisée 5-fold. Nous avons validé miRBoost sur chacune des cinq sous-échantillons de données humaines et inter-espèces avec les caractéristiques identifiées sur les quatre sous-échantillons restants. Puis nous avons sélectionné un ensemble de caractéristiques sur chaque ensemble de données d'humain et d'inter-espèces. miRBoost est ensuite évalué sur chacun des cinq sous-échantillons avec les caractéristiques sélectionnées. La Table 4.4 indique le nombre de caractéristiques sélectionnées dans chaque ensemble d'apprentissage validé et dans l'ensemble des jeux de données pour l'humain et les inter-espèces.

Espèces	Ens_1	Ens_2	Ens_3	Ens_4	Ens_5	*Tous*	*Communs*
Humain	26	19	30	27	27	27	12
Inter-espèces	26	28	25	33	21	27	12
Communs	11	5	14	13	13	14	3

TABLE 4.4 – Nombre de caractéristiques sélectionnées pour chaque ensemble d'apprentissage. Ens_i représente le i^{th} ensemble d'apprentissage, $i = 1, \ldots, 5$ dans la validation croisée, *Tous* représente tout l'ensemble d'apprentissage, et *Communs* représente les caractéristiques communes entre les ensembles de données.

4.3.5 Résultats

Nous avons comparé miRBoost à plusieurs outils bioinformatiques existants pour la classification des pré-miARNs : *CSHMM* [2] et *triplet-SVM* [211], qui ne prennent pas en compte le problème de déséquilibre des données ; *microPred* [11], *mirExplorer* [65], *MiPred* [87], et *HeteroMirPred* [112], qui traitent ce problème ; et *MIReNA* [124] qui n'utilise pas de méthode d'apprentissage automatique pour la classification.

Données utilisées pour l'apprentissage

Nous avons utilisé différents ensembles de données positives et négatives pour effectuer la validation croisée sur le génome de l'homme et sur des génomes de plusieurs espèces (inter-espèces). Les génomes des

eucaryotes contenant au moins 100 miARNs dans la base de données miRBase (version 18) [98] sont sélectionnés. Les pré-miARNs qui contiennent moins de 400 nucléotides sont considérés comme les données positives. Cela comprend 1 527 séquences pour l'homme et 18 122 pour les inter-espèces. Pour éviter le sur-apprentissage, nous avons enlevé les séquences qui ont une identité de plus de 97% avec les autres. Cela donne 982 pré-miARNs pour l'humain et 7 463 pour les inter-espèces.

Pour les données négatives, nous avons récupéré aléatoirement, à partir du serveur de NCBI [136], les régions exoniques de gènes codant pour des protéines dans les génomes sélectionnés. Nous avons utilisé *miRNAFold* pour prédire la structure en épingle à cheveux dans chaque séquence sélectionnée. Plusieurs contraintes sont appliquées à la prédiction de la structure. La structure en épingle à cheveux doit avoir une énergie libre $\Delta_G < -25,0$. La séquence contient moins de 150 nucléotides, tandis que son épingle à cheveux est formée avec au moins une tige exacte de plus de 5 nucléotides. En outre, au moins 90% des caractéristiques introduites dans *miRNAFold* doivent être satisfaites. Les structures en épingle à cheveux ainsi obtenues sont alors considérées comme les données négatives, comprenant 34 677 séquences d'humain et 177 133 séquences inter-espèces. Nous réduisons également la redondance des séquences à 9%, donnant ainsi 7 123 et 15 832 séquences qui ne sont pas des pré-miARNs pour l'homme et les inter-espèces respectivement.

Performance de la classification

Nous avons mesuré les performances de miRBoost en utilisant le validation croisée 5-fold sur les ensembles de données de l'humain et d'inter-espèces décrites ci-dessus. Dans cette validation croisée, nous avons comparé miRBoost à CSHMM, triplet-SVM, microPred et MiReNA. Les autres outils cités plus haut ne donnent pas la possibilité de ré-apprendre leurs modèles avec d'autres données, nous n'avons donc pas pu les considérer ici. Pour CSHMM, triplet-SVM et microPred, nous avons effectué une validation croisée 5-fold avec les mêmes ensembles d'apprentissage et de test que dans l'évaluation de miRBoost. Nous avons ensuite ré-appris leurs modèles pour nos ensembles d'apprentissage.

Nous avons ensuite comparé nos résultats avec ceux de ces méthodes, indépendamment sur chaque ensemble de données. Pour évaluer la performance de la classification, nous avons utilisé différentes mesures statistiques. Nous avons calculé la sensibilité SE, le coefficient de corrélation de Matthews MCC, la précision ACC, le F-score et la métrique géométrique g-mean définies comme suit :

- Sensibilité $SE = \frac{TP}{TP+FN}$, qui mesure la précision sur les échantillons positifs,
- Précision $ACC = \frac{TP+TN}{TP+TN+FP+FN}$, qui mesure le pourcentage d'échantillons bien classés,
- F-score $= \frac{2 \times TP}{2 \times TP + FP + FN}$, la moyenne harmonique de la sélectivité ($\frac{TP}{TP+FP}$) et de la sensibilité ($\frac{TP}{TP+FN}$),
- Le coefficient de corrélation de Matthews $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, où la valeur $+1$ représente une prédiction parfaite, tandis que -1 indique une prédiction inverse,
- g-mean $= \sqrt{SE \times SP}$, qui est généralement utilisée pour l'évaluation des classifieurs sur des données déséquilibrées,

tel que TP, FP, TN, FN représentent le nombre de vrai positifs, faux positifs, vrai négatifs et faux négatifs, respectivement.

Les résultats de la classification de miRBoost et des autres algorithmes sont présentés dans Table 4.5. Le meilleur score de chaque mesure est indiqué en caractères gras.

Comme on peut le voir, miRBoost donne la meilleure précision, F-score, MCC et g-mean pour chacun des ensembles de données pour l'humain et les inter-espèces. Pour toutes les mesures considérées, miRBoost donne toujours des scores supérieurs à 0.87 (supérieurs à 0.97 pour les inter-espèces). miRBoost est clairement meilleur que les méthodes ne tenant pas compte du problème de déséquilibre des données, à savoir

	Homme					Drosophile				
Logiciel	ACC	SE	F-score	MCC	g-mean	ACC	SE	F-score	MCC	g-mean
miRBoost	97.3%	87.0%	**0.89**	0.87	**0.93**	**98.9%**	**98.2%**	**0.98**	**0.97**	**0.99**
CSHMM	93.7%	56.4%	0.68	0.67	0.75	82.3%	54.3%	0.66	0.58	0.72
triplet-SVM	95.2%	72.8%	0.79	0.76	0.85	86.9%	74.4%	0.78	0.69	0.83
microPred	**97.5%**	83.6%	**0.89**	**0.88**	0.91	95.9%	91.0%	0.93	0.90	0.95
MIReNA	92.0%	82.7%	0.72	0.68	0.88	88.6%	78.0%	0.81	0.73	0.85

TABLE 4.5 – Performance de miRBoost en comparaison avec d'autres méthodes de classification de pré-miARNs sur les données de l'humain et de la drosophile.

CSHMM et triplet-SVM, et meilleur que MIReNA. Concernant *microPred*, qui traite donc les données déséquilibrées, celui-ci a des résultats comparables pour l'humain, mais moins bons que ceux de miRBoost pour les données inter-espèces.

Sensibilité de la prédiction sur de nouvelles séquences

Nous avons évalué la sensibilité de miRBoost et des autres méthodes sur 3 320 nouvelles séquences de pré-miARNs introduites dans les versions 19 et 20 de la base de données miRBase (miRBoost a été initialement appris sur les données de la version 18 de miRBase). Pour miRBoost, nous avons utilisé le modèle appris sur l'ensemble des données inter-espèces avec ses 27 caractéristiques sélectionnées. De même, les modèles de CSHMM, triplet-SVM et microPred ont également été appris sur cet ensemble de données. Pour MiPred, HeteroMirPred et mirExplorer, nous avons toujours utilisé leurs modèles fournis. Les résultats des tests sont donnés dans Table 4.6. miRBoost prédit le plus grand nombre de pré-miARNs et réalise ainsi la plus grande sensibilité par rapport aux autres algorithmes. Cela confirme que notre méthode peut être considéré comme plus performant par rapport aux différentes méthodes de classification de pré-miARNs déjà existantes dans la littérature.

Logiciel	TP	SE (%)	$Temps$
miRBoost	**3 257**	**98.10**	6m43s
CSHMM	1 960	59.04	31m29s
triplet-SVM	2 462	74.16	39s
microPred	3 094	93.19	43h44m39s
MIReNA	2 613	78.70	39s
HeteroMirPred	2 866	86.33	7h40m2s
MiPred	1 876	56.51	9h58m46s
mirExplorer	2 926	88.13	11m38s

TABLE 4.6 – Comparaison de miRBoost avec les autres méthodes dans la prédiction de 3 320 nouveaux pré-miARNs extraits des versions 19 et 20 de miRBase. Sont donnés ici le nombre de pré-miARNs vrais positifs prédits (TP), la sensibilité obtenue (SE) et le temps d'exécution (Temps). Le temps d'exécution de mirExplorer est mis en gris car effectué sur une machine différente.

Temps d'exécution

Avec les grands volumes de données générés par les nouvelles technologies de séquençage, le temps d'exécution devient un facteur important dans l'évaluation des outils de prédiction. miRBoost présente un temps d'exécution raisonnable par rapport aux autres outils, comme on peut le voir dans la Table 4.6.

Une grande partie du temps d'exécution de miRBoost est pour quantifier les caractéristiques des séquences considérées, c'est à dire pour calculer les valeurs numériques des différentes caractéristiques, nécessaires comme entrée pour les classifieurs SVM. Nous utilisons miRNAFold pour replier la séquence en structure d'hairpin, puis nous générons dans un second temps des valeurs de caractéristiques pour la classification. Dans triplet-SVM et MiRena, qui sont les deux méthodes les plus rapides, un filtre est utilisé en amont, c'est à dire avant la quantification des caractéristiques, afin de rejeter rapidement les séquences qui ne satisfont pas à certaines contraintes et donc les classer comme échantillons négatifs, leur permettant ainsi de gagner un temps non négligeable.

Tous les tests ont été effectués sur une machine Linux avec 24-core Intel Xeon X5680 de 3.33GHz et 20Gb de RAM, à l'exception de *mirExplorer*, qui a été exécuté sur une machine Windows machine avec Core 2 Duo Intel E8400 de 3.00GHz et 4Gb de RAM (pour cause de non portabilité sur une machine Linux).

4.3.6 Conclusion

Nous avons développé un algorithme rapide et efficace pour la classification des pré-miARNs. Basé sur de l'apprentissage automatique, il traite le problème des données déséquilibrées par un boosting de classifieurs SVM affaiblis. Notre méthode est très précise, avec plus de 97% de précision, 0.89 F-score, 0.87 MCC, et 0.93 g-mean. Elle montre aussi plus de 98% de sensibilité dans la prédiction de nouveaux pré-miARNs. Les résultats de classification, ainsi que le temps d'exécution, sont avantageusement comparables à ceux donnés par les méthodes existantes. Un tel outil pourrait être utile pour une prédiction de pré-miARNs à l'échelle du génome.

La performance de miRBoost peut être améliorée en utilisant le boosting pour la sélection des caractéristiques [157, 26]. En outre, d'autres types de SVM qui gèrent mieux les données déséquilibrées [3] pourraient également avoir une contribution considérable dans le boosting pour prendre en compte la question du déséquilibre des données.

Actuellement, nous travaillons sur l'intégration de miRBoost à miRNAFold, afin de proposer un algorithme qui permette de rechercher des miARNs dans des génomes entiers en des temps et avec des sélectivités acceptables. Comme nous l'avons vu plus haut, miRNAFold, qui a surtout l'avantage d'être très rapide, donne une sensibilité élevée (autour de 90%), mais a le désavantage d'avoir une sélectivité faible. En d'autres termes, le nombre de miARNs prédits est trop élevé. En intégrant miRBoost à miRNAFold, la sélectivité de miRNAFold sera améliorée, mais au détriment du temps d'exécution. Il s'agira donc d'avoir une nouvelle version de miRNAFold qui donnerait de meilleurs sélectivités que miRNAFold tout en restant compétitif d'un point de vue temps d'exécution.

Chapitre 5

Prédiction de petits ARNs non-codants en lien avec les éléments transposables

5.1 Introduction

Des études récentes ont montré que les génomes entiers des eucaryotes supérieurs sont transcrits alors que les gènes ne représentent que quelques pourcents de ces génomes. Les régions non-géniques sont principalement composées par des ARNs non-codants et par des éléments transposables (ET ou TE), qui représentent une partie importante de nombreux génomes eucaryotes. Par exemple, environ 50% du génome humain est dérivé à partir de séquences d'éléments transposables [30]. Les éléments transposables sont présents dans presque tous les génomes qui ont été étudiés à ce jour et dans certains cas, représentent la majorité du génome.

5.1.1 Problème d'annotation des ARNncs liés aux éléments transposables

Des études bioinformatiques récentes montrent que certains pré-miARNs partagent leurs séquences ou une partie importante de leurs séquences avec des TEs [143, 39, 80, 126]. Ces pré-miARNs, annotés dans miRBase, sont appelés des (pré-)miARNs TE dérivés [143]. Toutefois, parmi ces pré-miARNs, certains d'entre eux présentent un grand nombre d'occurrences dans le génome et correspondent entièrement à des TEs. Par exemple, les séquences des pré-miARNs HSA-MIR-548a1 et HSA-MIR-548a2 correspondent exactement à MADE1 dans la base de données RepBase, base de données recensant les TEs connus. Yan et al. ont par exemple montré expérimentalement que mir441 et miARN446 correspondent à des siARNs [212].

Les ETs non-autonomes courts et certains précurseurs d'ARNs non codants tels que les pré-miARNs sont caractérisés par une taille similaire et une structure secondaire en épingle à cheveux (voir Figure 5.1). Par exemple, la séquence MITE Hsmar1 de l'humain est de 80 nt de long et elle forme une structure secondaire en épingle à cheveux [90]. La transcription de ces MITE par l'ARN polymérase II peut conduire à la synthèse de rasiARNs ("repeat associated small interfering RNAs") et à des ARNs piwi. Ces petits ARNs sont de taille similaire aux miARNs [212, 99, 197]. En outre, les rasiARNs déclenchent la régulation post-transcriptionnelle utilisant des protéines de Dicer-like comme les miARNs [212, 197].

Des critères pour annoter les microARNs ont été proposés en 2003 puis ont évolué pour prendre en compte les données produites par l'utilisation des technologies de séquençage massivement parallèles [98]. Cependant, certaines études montrent que certains gènes de miARNs sont mal annotés. Par exemple, Yan *et al.* ont montré expérimentalement que OSA-MIR441 et OSA-MIR446 correspondent à de petits ARNs interférents [212]. Langenberger et ses collègues ont montré que les snoARNs étaient souvent mal annotés comme miARNs [104]. Dans un autre exemple, un gène de miARN est entièrement inclus dans un TE, ce qui est le

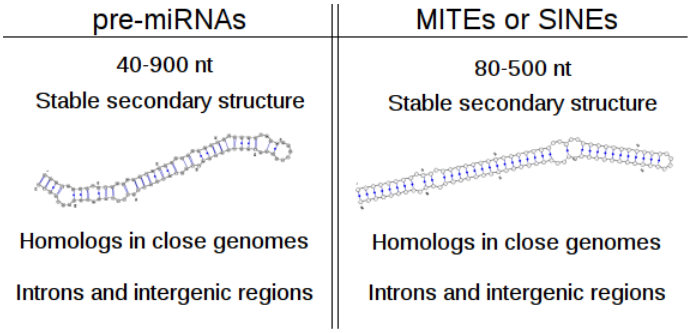

FIGURE 5.1 – Les TEs non-autonomes et les gènes répétés inversés partagent des caractéristiques biologiques, telle que la structure en épingle à cheveux. Les structures indiquées ici correspondent au pré-miARN CEL-LET-7 dans le nématode et à une occurrence d'un ET MADE1 dans le génome humain.

cas du HSA-MIR-1255a présent sur le chromosome 4 [62]. Ce lieu correspond également au MITE Tigger1. La même situation se retrouve pour les 58 membres de la famille HSA-MIR-548.

Les utilisateurs ont de ce fait besoin d'un outil pour les aider à annoter les petites séquences d'ARNnc liées à des TEs, et déterminer s'il s'agit d'un ARNnc dérivé d'un ET ou s'il s'agit d'une éventuelle mal-annotation d'un ET en ARNnc.

5.1.2 Les piARNs, petits ARNncs dérivés des éléments transposables et encore mal connus

Les piARNs, petits ARNs récemment découverts (voir Section 2.2.4 et Section 2.3.3), ont pour rôle principal la protection du génome contre l'invasion des ETs, en s'associant à eux pour les dégrader. Ce sont des séquences qui dérivent a priori d'ETs, car ressemblant fortement à ces derniers : il sont riches en séquences répétées, et apparaissent sous forme de clusters sur le génome [113].

L'identification et la caractérisation de piARNs de mammifères a été en grande partie effectuée par des approches expérimentales, qui combinent l'isolement des séquences interagissant avec des protéines PIWI et/ou le "deep sequencing" de courtes séquences d'ARNs dans les lignées germinales [56, 7, 201]. Bien que cette méthodologie est apparu productive, elle ne peut couvrir de manière exhaustive l'ensemble du répertoire des molécules de piARNs dans un organisme étudié. Disposer d'outils bioinformatiques pour aider à les identifier peut donc être un atout non négligeable, d'autant plus l'intérêt pour cette classe d'ARNncs encore mal connue sera sans nul doute croissant, de plus de plus d'études suggérant de nouvelles fonctions biologiques à ces ARNncs [147, 162] et leur implication dans des maladies tel que le cancer [128].

Contrairement aux miARNs, on ne connaît à ce jour quasiment pas de caractéristiques spécifiques aux piARNs, ni de de structure secondaire, à part leur occurrence sous forme de clusters sur le génome, et la présence d'un uridine (U) en première base en 5' chez la plupart d'entre eux [113]. Ils ne sont a priori pas conservés entre espèces, ni même au sein d'une même espèce, rendant ainsi leur identification par des méthodes *in silico* très difficiles. Il n'existe d'ailleurs quasiment pas d'outils permettant de les prédire. Une seule méthode a été proposée pour prédire si une séquence est un piARN ou non [215]. Elle effectue une classification linéaire basée sur la détermination de différents motifs k-mer dans les séquences considérées. Tous les motifs de taille 1 à 5 nt sont considérés, dont 4 motifs 1-mer (A, C, G et T), 16 motifs 2-mer, 64 motifs 3-mer, 256 motifs 4-mer et 1 024 motifs 5-mer. Un total de 1 364 motifs sont obtenus, et utilisés pour classer une séquence en un piRNA ou non piARN. Il existe par ailleurs deux méthodes basées sur des approches de clustering, proTRAC ([161]) et piClust ([89]), pour l'identification *in silico* de clusters de piARNs à partir de données RNAseq. L'algorithme proTRAC est basé sur une analyse probabiliste statistique. Il identifie les clusters sur la base d'écarts importants à partir d'une distribution uniforme des piRNAs, utilisant différentes informations, dont la densité du read, l'asymétrie du brin, la fréquence de piRNAs avec

U à la première position de la séquence, ou A à la position 10. En revanche, l'algorithme piClust utilise une approche de classification basée sur la densité sans hypothèse sur aucune distribution paramétrique, et considérant la distance réelle entre les reads pour la détermination de clustering.

5.1.3 Notre contribution

Avec Sébastien Tempel et en collaboration avec Nicolas Pollet d'ISSB, nous avons développé une méthode automatique appelée *ncRNAclassifier* pour déterminer la catégorie d'un petit ARNnc donné selon sa similarité (ou non) avec une séquence TE, et en se basant sur le pourcentage de TE dans leur séquence, et leur dispersion dans le génome. Nous avons analysé grâce à *ncRNAclassifier* les pré-miARNs de miRBase de plusieurs espèces : l'homme, la souris, le nématode, le sea squirt, le rat et la grenouille. Nous avons constaté que des centaines de pré-miARNs de l'humain et de la souris, et certains de la grenouille, du nématode, du rat et du sea squirt peuvent être classés comme étant dérivés de TE. Nous avons également observé de nombreux exemples de pré-miARNs correspondant en très grande partie à des TEs et qui devraient donc être ré-annotés comme TEs.

Grâce à *ncRNAclassifier*, on peut vérifier très rapidement si un candidat pré-miARN est un ET-dérivé ou un ET. Il faut entre 30 secondes à 1 minute pour traiter une séquence pré-miARN. Une première version de ce travail a été publié dans la conférence nationale JOBIM en 2011 [184] puis une version étendue a été publié dans la revue BMC Bioinformatics [183].

Très récemment, en collaboration avec David Israeli du Généthon et Farida Zehraoui, et dans le cadre du stage de M2 de Jocelyn Brayet, nous avons développé une première version d'un algorithme pour la prédiction des piARNs, dont la particularité est d'être modulaire, et donc adaptatif et extensible. Il s'agit d'une méthode d'apprentissage automatique supervisée, basée sur la combinaison de SVM (Support Vector Machine) et d'une fusion de plusieurs noyaux (similarités), chaque noyau représentant une caractéristique (ou éventuellement plusieurs caractéristiques de même type) des piARNs. Les piARNs étant encore mal connus, et étant non conservés entre espèces, l'intérêt de notre algorithme, appelé *piRPred*, est d'une part de pouvoir intégrer de nouvelles caractéristiques des piARNs (au fur et à mesure qu'on en découvre), et d'autre part de pouvoir adapter le traitement à l'espèce considérée. Nous avons testé *piRPred* sur l'Humain et la Drosophile, et les premiers résultats obtenus sont très prometteurs. Malgré le peu de caractéristiques utilisées dans cette première version de l'algorithme, la sensibilité et la sélectivité des résultats de prédiction sont autour de 80%. Ce travail préliminaire sur la prédiction des piARNs a fait l'objet d'une soumission à la conférence européenne de bioinformatique ECCB'14.

5.2 *ncRNAclassifier* : Identification des ARNs non-codants dérivés d'éléments transposables

Nous avons développé une méthode automatique appelé *ncRNAclassifier* permettant de classer les séquences de précurseurs d'ARNnc selon leur similarité avec des séquences TE, et de prédire d'éventuelles mal-annotations de certains ARNnc, car correspondant à des TEs.

5.2.1 Notre approche

Notre méthode est basée sur le postulat qu'un pré-ARNnc qui a plusieurs occurrences répandues dans le génome a une forte probabilité d'être soit dérivé d'un TE ou d'être mal-annoté comme étant un pré-ARNnc alors qu'il est un TE. La première étape de *ncRNAclassifier* est de calculer le nombre d'occurrences du candidat, le nombre de chromosomes où apparaissent les différentes occurrences et la distance entre les occurrences. La seconde étape calcule ensuite une séquence consensus à partir des dix occurrences les plus similaires à la séquence ARNnc. Enfin, la dernière étape vérifie si la séquence consensus correspond à un TE dans la base de données RepBase.

Etant donné un pré-ARNc, nous le classons ainsi dans l'une des trois catégories suivantes :

- précurseur dont la séquence est dépouillée de toute séquences TE-dérivé et non répétée ni dispersée dans une large mesure dans le génome: "véritable" (*bona fide*) pré-ARNnc (ou gène ARNnc).
- précurseur dont la séquence correspond à une petite partie d'une séquence connue TE et/ou qui est répétée et dispersée dans le génome : ARNnc TE-dérivée.
- précurseur dont la séquence correspond à une grande partie d'une séquence TE connue, soit déjà annotée en tant que telle ou identifiée par notre méthode : ARNnc mal annoté.

5.2.2 Description de la méthode

Notre méthode *ncRNAclassifier* est présentée en Figure 5.2. Elle prend en entrée la séquence d'un ARNnc candidat et en sortie spécifie s'il s'agit (i) d'un ARNc sans lien avec un ET, (ii) d'un ARN ET-dérivé ou (iii) d'un ARNnc mal annoté. La méthode comporte plusieurs étapes :

- Dans la première étape nous étudions la distribution des occurrences de la séquence donnée en entrée en utilisant BLAT [94] du navigateur UCSC Genome [53]. BLAT retourne les occurrences de séquences ("hits") qui sont similaires à la séquence du précurseur donné, et les chromosomes où ils apparaissent. On en déduit le nombre de "hits" similaires, qui sont des hits dont la similitude avec le candidat est égale ou supérieure à 80% et dont la taille est comprise entre 80% et 120% de la taille du précurseur. Ces seuils sont aussi utilisés dans [152]. Ensuite, nous calculons le nombre de chromosomes contenant ces hits similaires.
- Dans la deuxième étape, la séquence entourant chaque hit est déduite : 100 nt vers la gauche et vers la droite. Nous avons besoin de ces bits supplémentaires de séquence parce que la taille de certains précurseurs ARNnc pourrait être trop courte pour évaluer les similitudes possibles avec des éléments transposables connus (les pré-miARNs humains varient par exemple entre 60 et 140 nt [98]). Les séquences obtenues sont ensuite alignées en utilisant ClustalW [105] et une séquence consensus est créée. Le consensus de nucléotides en position *i* correspond au nucléotide présent au moins cinq fois dans l'alignement à la même position, sinon nous mettons le caractère N.
- Dans la troisième étape, nous utilisons CENSOR [91] pour comparer la séquence consensus précédemment créée avec la base de données de ETs RepBase [90].
- Dans la quatrième étape, qui est optionnelle et activée lorsque l'utilisateur entre les coordonnées génomiques de l'ARNnc, l'annotation de RepeatMasker de UCSC Génome est vérifiée. Les résultats de CENSOR et les résultats de RepeatMasker sont ensuite comparés pour garder le plus grand fragment ET.

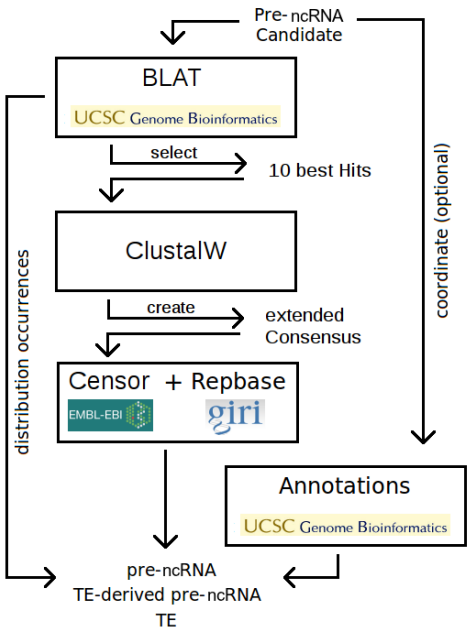

FIGURE 5.2 – La méthode *ncRNAclassifier* : étant donnée un ARNnc, la première étape consiste à appliquer BLAT du navigateur UCSC Genome et d'obtenir les dix hits les plus similaires ; la deuxième étape consiste ensuite à aligner ces résultats par ClustalW afin d'obtenir une séquence consensus qui est prolongée ; enfin la dernière étape consiste à comparer la séquence consensus prolongée avec les séquences de la base de données de ETs RepBase en utilisant CENSOR de EBI.

- La cinquième étape procède à la classification. Nous distinguons deux cas. Le premier cas est quand un segment de 24 nt (taille d'un mi- ou siARN mature [10]) non lié à un ET peut être trouvé. Ainsi, un petit ARN mature peut être généré à partir du précurseur, et peut être relié à un ARNm cible dépouillé de toute séquence ET. Nous appelons cela un pré-ARNnc ET-dérivé. Dans le second cas, un tel segment ne peut être trouvé. Ainsi, un petit ARN mature généré à partir d'un tel précurseur se lierait à un ARNm cible à travers une séquence ET. Nous considérons qu'il s'agit ici d'un ET mal annoté en ARNnc.
- Enfin, notre méthode utilise la distribution des occurrences et la taille de la séquence TE afin de classer le candidat pré-ARNnc. Sur la base de ces deux caractéristiques, *ncRNAclassifier* classe le candidat selon les règles suivantes :
  - une occurrence, pas de TE reconnaissable $\Rightarrow$ *bona fide* pré-ARNnc
  - plus de 20 occurrences, pas de TE reconnaissable $\Rightarrow$ pré-ARNnc TE-dérivé
  - des occurrences sur six chromosomes ou plus, pas de TE reconnaissable $\Rightarrow$ pré-ARNnc TE-dérivé
  - une occurrence ou plus, TE reconnaissable et segment non lié à un TE $\geq$ 24 nt $\Rightarrow$ pré-ARNnc TE-dérivé
  - une occurrence ou plus, TE reconnaissable et segment non lié à un TE $<$ 24 nt $\Rightarrow$ TE

  Notons que l'utilisateur peut choisir les seuils utilisés dans la classification des précurseurs ARNnc, à savoir le nombre minimal de hits similaires et le nombre minimal de chromosomes.

5.2.3 Résultats

Analyse par *ncRNAclassifier* des pré-miARNs de miRBase

Nous avons utilisé *ncRNAclassifier* pour l'analyse des pré-miARNs de miRBase [98] sur six génomes : frog (*Xenopus tropicalis*), humain (*Homo sapiens*), souris (*Mus musculus*), nematode (*Caenorhabditis elegans*), rat (*Rattus norvegicus*) et sea squirt (*Ciona intestinalis*). Les résultats obtenus concernant le nombre de

pré-miARNs TE-dérivés et le nombre de pré-miARNs mal-annotés sont donnés en Table 5.1.

	Total de pré-miARNs	Mal-annotés	TE-dérivés
Grenouille	182	1	3
Humain	1037	235	152
Souris	542	68	110
Nématode	200	2	5
Rat	359	28	21
Sea squirt	310	2	19

TABLE 5.1 – Nombre de pré-miARNs de miRBase qui sont TE-dérivés ou mal-annotés pour les génomes de grenouille, humain, souris, nématode, rat, et sea squirt.

Comme nous pouvons le voir, ncRNAclassifier a mis en évidence un nombre non-négligeable de cas de mal-annotation et des relations évidentes avec des TEs dans les six génomes étudiés. Nous avons par exemple noté qu'un pré-miARN chez le sea squirt correspond complètement à l'élément transposable $HAT5N_{CI}$. En outre, nous avons observé que huit pré-miARNs mal-annotées chez le rat correspondent complètement (à 100%) à des TEs. Par ailleurs, certains pré-miARNs TE-dérivés ou mal-annotés dans les différents génomes sont composés de deux ou plusieurs fragments TE distincts.

Certaines études ont rapporté l'identification de pré-miARNs TE-dérivés [103, 151, 152]. En outre, Jordan *et al.* ont montré que six pré-miARNs humains (HSA -MIR -548) correspondent à des TE [150]. Ils les ont appelés "miARNs TE-dérivés". Par ailleurs, la base de données microTranspoGene liste des pré-miARNs "TE-dérivés" de miRBase [114]. Cependant, cette base de données est basée sur la version 10.0 de miRBase et il n'y a aucun nouveau miARN TE-dérivé depuis 2007. Notre méthode automatique reproduit les résultats obtenus dans [150, 103, 151, 152, 171]. *ncRNAclassifier* identifie la plupart des miARNs TE-dérivés décrits dans ces études, ainsi que ceux énumérés dans la base de données microTranspoGene. Nous avons identifié avec *ncRNAclassifier* respectivement 138, 99, 4, 21 et 14 pré-miARNs TE dérivés chez l'homme, la souris, le nématode, le rat et le sea squirt, dont 108, 88, 3, 21 et 13 non identifiés dans la littérature. Nous avons également identifié 1, 235, 68, 2, 28 et 2 pré-miARNs mal-annotés chez la grenouille, l'humain, la souris, le nématode, le rat et le sea squirt, dont 1, 194, 57, 2, 28 et 2 n'ont pas été identifiés auparavant dans la littérature. Enfin, les six pré-miARNs humains identifiés par Jordan *et al.* comme des "miARNs TE-dérivés" ont été identifiés par *ncRNAclassifier* comme des pré-miARNs mal-annotés.

Distribution des occurrences de pré-miARNs dans les génomes et leur lien à des TEs

Nous avons examiné la distribution des occurrences de pré-miARNs dans les six génomes considérés, en fonction des trois catégories définies par *ncRNAclassifier* (voir Figure 5.3). Nous avons trouvé une corrélation positive entre les pré-miARNs mal-annotés ou TE-dérivés et le nombre de hits similaires. Les pré-miARNs mal annotés ont été caractérisés par le plus grand nombre de hits et la plus haute dispersion sur les chromosomes. Les pré-miARNs TE-dérivés ont été caractérisés par moins de hits similaires sur les chromosomes par rapport aux pré-miARNs mal-annotés, et les pré-miARNs sans séquence de TE ont le plus petit nombre de hits. Ce résultat est particulièrement remarquable sur les génomes humain et souris. Nous avons constaté que la majorité des pré-miARNs qui ne correspondent pas à des TEs connus ont seulement un hit similaire. Seulement 36 pré-miARNs parmi un total de 3 276 pré-miARNs analysés dans les six espèces (1,1%) ont plus de 20 hits similaires ou sont présents dans plus de 6 chromosomes mais classés comme sans rapport avec des TEs par ncRNAclassifier. Dans le cas des génomes de grenouille, nématode, rat et sea squirt, nous avons observé que certains pré-miARNs qui ne sont pas identifiés comme correspondant à des TEs mais ayant de nombreux hits dans plusieurs chromosomes ont en fait seulement deux occurrences sur deux chromosomes. C'est par exemple le cas de 29 pré-miARNs de grenouille parmi les 32 correspondant à des TEs.

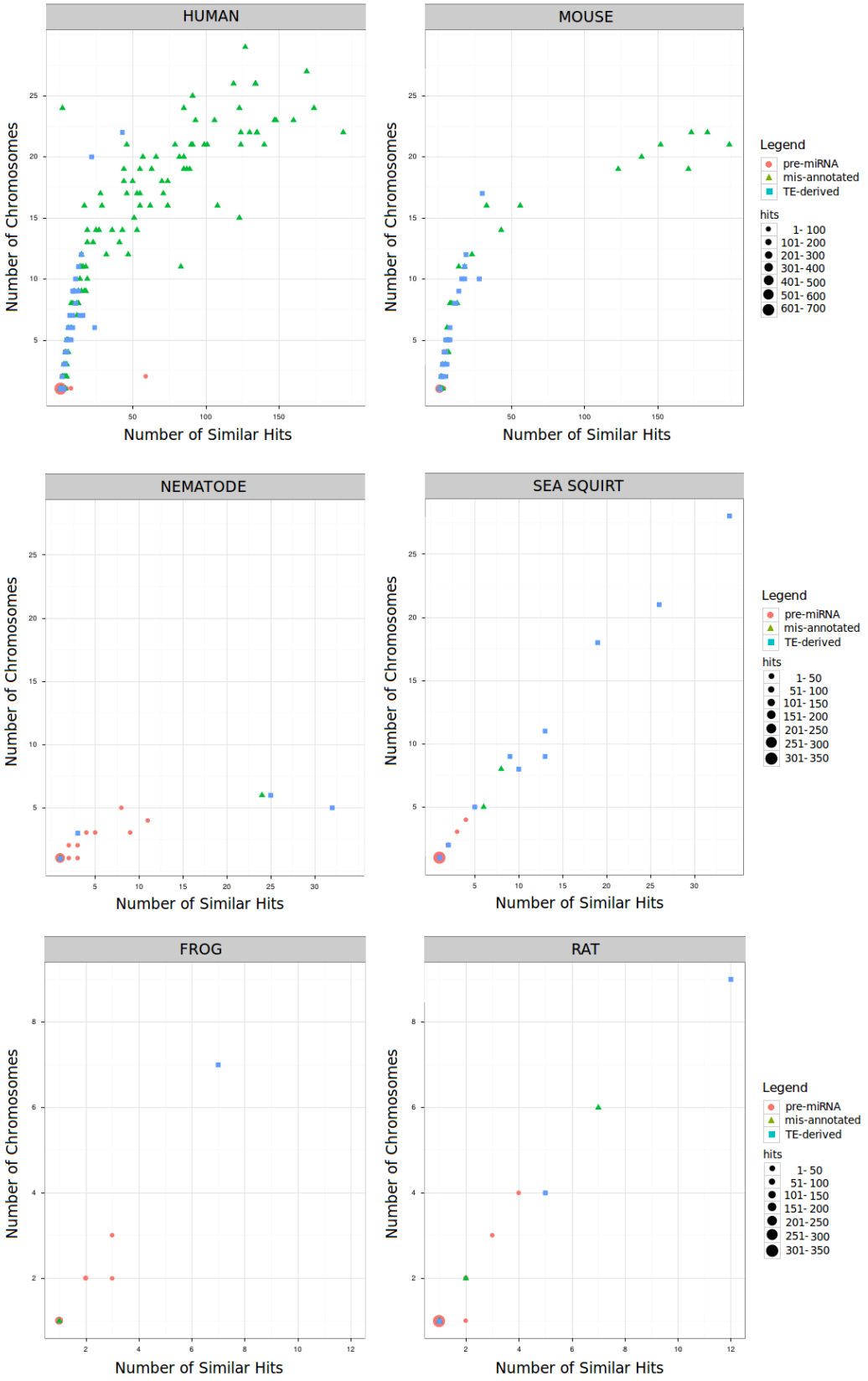

FIGURE 5.3 – Distribution des hits de pré-miARNs dans les génomes de grenouille, humain, souris, nématode, rat et sea squirt. En rouge : les pré-miARNs identifiés par *ncRNAclassifier* comme des pré-miARNs ne correspondant pas à des TEs. En bleu : les pré-miARNs identifiés comme des TE-dérivés. En vert : les pré-miARNs identifiés comme des TEs. La taille des points dépend du nombre de pré-miARNs considérés.

5.2.4 Conclusion

Parmi les pré-miARNs de miRBase, nous avons identifié des centaines de cas de mal-annotation où des TEs sont considérés comme des pré-miARNs : 235 cas concernant le génome humain et 68 cas pour le génome de la souris, avec respectivement 194 et 57 cas qui ne sont pas mentionnés dans la littérature.

Grâce à ncRNAclassifier, on peut vérifier très rapidement si une séquence ARNnc avec une structure en épingle à cheveux donnée correspond à une séquence TE. Il faut entre 30 secondes à 1 minute pour traiter une séquence, en fonction du nombre d'occurrences dans UCSC et de l'accès à RepBase à EBI.

5.3 *piRPred* : Prédiction de piARNs

5.3.1 Problématique de l'identification des piARNs et approche développée

Comme nous l'avons dit précédemment, il n'existe actuellement quasiment pas de méthodes bioinformatiques pour la prédiction des piARNs. Ces ARNs sont en effet difficiles à prédire, et la raison principale est leur manque de conservation entre espèces et au sein même d'une espèce, contrairement aux miARNs. On ne leur connaît pas de structure secondaire ni de motifs particuliers au niveau des séquences, à l'exception de l'occurrence d'une uridine en première base [113]. Par contre, il est connu et admis leur occurrence en clusters sur le génome, pouvant s'étendre jusqu'à 200Kb chez certaines espèces telles que la drosophile [113, 19]. Rosenkranz et Zischler en 2012 puis Jung *et al.* en 2014 ont ainsi exploité cette caractéristique pour développer respectivement les méthodes proTRAC [161] et piClust [89] pour prédire des clusters de piARNs à partir de données de séquençage. Par ailleurs, Zang *et al.* ont montré qu'un ensemble de motifs (k-mer) sont différentiellement fréquents dans les séquences piARNs et non-piARNs et ont exploité cette particularité pour classer des séquences données en piARNs ou non-piARNs [215].

Nous avons donc proposé une nouvelle méthode de classification des piARNs basée sur de l'apprentissage automatique supervisée. Nous avons voulu dans un premier temps combiner dans un même algorithme les caractéristiques exploitées séparément par les méthodes ci-dessus (k-mer, proTRAC et piClust), à savoir l'occurrence en clusters et les fréquences des motifs k-mer. Nous avons bien sûr considéré la caractéristique de l'uridine en première position, exploitée aussi dans les autres algorithmes. Et nous avons également utilisé une nouvelle caractéristique, non encore exploitée, et qui concerne la présence des clusters de piARNs dans des régions proches des télomères et du centromère, caractéristique observée chez la drosophile par Brennecke *et al.* [19].

Les différentes caractéristiques que nous avons considérées sont hétérogènes et de différents types. De plus, certaines d'entre elles sont spécifiques à certaines espèces. Pour prendre en compte cette particularité du contexte, nous avons développé une méthode modulaire, basée sur la définition de plusieurs noyaux, chaque noyau représentant un type de caractéristique. Nous avons ainsi une méthode adaptative et extensible : on peut adapter l'utilisation de certains noyaux ou d'autres selon l'espèce considérée, et on peut intégrer dans le système de nouvelles caractéristiques par une simple définition de nouveaux noyaux.

Dans la version actuelle de notre algorithme, nous considérons donc les caractéristiques suivantes :

1. La fréquence de certains motifs k-mer.
2. La présence d'une base d'uridine à la première position de la séquence.
3. La distance par rapport aux régions centromiques et télomériques du chromosome.
4. L'apparition de piARNs en clusters sur le génome.

Nous définissons trois noyaux : un noyau représentant les deux premières caractéristiques et deux noyaux représentant la troisième et la quatrième respectivement. Chaque noyau est une matrice de similarité carrée de taille $N \times N$, N étant la taille de l'ensemble des données d'apprentissage (comprenant les échantillons

positifs et négatifs). Nous construisons pour chaque séquence un vecteur (ou une matrice) représentant la caractéristique. Un noyau gaussien est ensuite construit à partir de ces vecteurs de la façon suivante : $k(x, y) = \exp^{-\gamma d(x,y)^2}$, où $d(x, y)$ représente la distance entre les vecteurs représentant les séquences, et γ un paramètre du noyau gaussien qu'on a estimé grâce à la méthode décrite dans ([207]) et qui consiste à calculer les distances entre classes dans l'espace des caractéristiques.

Pour effectuer la classification, nous calculons ensuite la moyenne des noyaux puis utilisons les SVM (Support Vector Machines).

5.3.2 Description des noyaux développés

Motifs k-mer et position de l'uridine Les k-mer se réfèrent à des k-tuples spécifiques d'acides nucléiques ou d'acides aminés qui peuvent être utilisés pour identifier certaines régions dans les biomolécules. Pour caractériser les séquences de piARNs, nous considérons des chaînes de k-mer, comme cela est fait par Zang *et al.* dans [215]. Nous utilisons les résultats obtenus par Zhang *et al.*, qui trouvent que 32 chaînes de k-mer (2 4-mer et 30 5-mer) sont différentiellement représentés entre les séquences de piARNs et non-piARNs. Nous avons donc calculé, pour chaque séquence, un vecteur contenant les fréquences de ces 32 k-mer dans la séquence.

Dans ce noyau nous avons ajouté l'information sur la présence ou l'absence d'une base d'uridine à la première position de la séquence. Nous considérons cette caractéristique des piARNs comme une caractéristique de l'apprentissage et non un filtre pour éviter d'éliminer les séquences qui ne présentent pas cette caractéristique. En effet, tous les piARNs ne présentent pas cette caractéristique. Nous avons analysé les séquences de piARNs de l'homme et de la drosophile disponibles dans piRNABank [102, 153], et respectivement 79,68% et 65,93% des piARNs de l'humain et de la drosophile ont une uridine en première position. Chaque séquence est alors représentée par un vecteur de 33 dimensions : la première dimension représente les informations sur la base d'uridine et les 32 autres dimensions représentent les fréquences des k-mer. Nous calculons ensuite un noyau gaussien en utilisant ces vecteurs.

Distances par rapport aux régions péricentromériques et subtélomériques Le second noyau correspond à la distance de la séquence par rapport aux régions péricentromériques et subtélomériques du chromosome. Nous avons construit un vecteur de caractéristiques de 4 dimensions qui représente la distance de la séquence à chacune de ces régions dans chaque chromosome (voir Figure 5.4) : la distance avec le premier télomère (t1), la distance avec le deuxième télomère (t2), la distance avec un côté du centromère (c1) et la distance avec l'autre côté du centromère (c2).

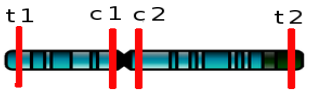

FIGURE 5.4 – Un chromosome avec les régions télomériques and centromérique.

Lorsque la séquence se trouve dans une région télomérique ou centromérique, la valeur de la distance est l'infini, ainsi que lorsque la séquence n'est pas sur le chromosome. Lorsqu'une séquence apparaît dans différents chromosomes sur le génome, la valeur minimale pour chacune des quatre distances est considérée. Le noyau gaussien est alors calculé avec ces valeurs minimales.

Clusters de piARNs en utilisant les k-plus proches voisins Pour tenir compte de la présence de clusters de piARNs de manière supervisée, nous avons construit un noyau qui prend en compte les voisins de chaque séquence dans le génome. Nous recherchons les k-plus proches voisins de chaque séquence, puis une matrice

de $(k+1)\times(k+1)$ contenant les distances entre toutes les séquences (la séquence cible et ses k-plus proches voisins) est construite. Nous calculons ensuite les distances de Frobenius entre les matrices obtenues. Un noyau gaussien est ensuite calculé en utilisant ces distances.

La valeur de k dépend du nombre de piARNs contenus dans un cluster. Cette valeur est très variable. La taille des clusters varie entre deux et plusieurs centaines de piARNs [56]. Nous avons fixé ce paramètre à 4 (voir Section 5.3.3)), mais peut être modifié par l'utilisateur.

5.3.3 Résultats

Données d'apprentissage

Nous utilisons des ensembles différents de données positives et négatives pour effectuer la validation croisée et les tests de prédiction sur les espèces *Humain* et *Drosophile*. Les données positives sont prises de piRNABank [102, 153]. piRNABank contient actuellement 23 439 et 22 336 séquences de piARNs sans redondance chez l'homme et la drosophile respectivement. Nous avons construit l'ensemble des données négatives avec des séquences non redondantes de plusieurs types :

- Séquences de taille entre 25 et 33 nucléotides correspondant à la partie 5' des séquences d'ARNt, qui sont prises d'une base de données de l'ARNt [64].
- Séquences correspondant à des miARNs matures et prises de miRBase [130].
- Séquences de taille comprise entre 25 et 33 nucléotides choisies aléatoirement parmi les régions exoniques de gènes codant pour des protéines récupérées de la base de données Ensembl75 via Biomart [14].

L'ensemble des données négatives de l'humain et de la drosophile contiennent respectivement 59 947 et 16 243 séquences non redondantes, composées de 590 et 301 séquences provenant d'ARNt, de 2 576 et 698 séquences de miARNs matures et de 56 781 et 15 244 séquences de régions exoniques.

Nous avons choisi aléatoirement pour l'homme et la drosophile un échantillon d'apprentissage de 7 500 (respectivement 1 300) séquences de l'ensemble des données positives et 7 500 (respectivement 1 300) séquences de l'ensemble des données négatives. Nous avons également choisi de la même manière 200 séquences positives et négatives (pour l'homme et la drosophile) autres que celles utilisées dans l'étape d'apprentissage, afin de tester notre algorithme sur la classification de nouvelles séquences.

Enfin, pour chaque séquence, nous avons les informations suivantes : le nom (id), la séquence nucléotidique, le brin («+» ou «-»), le nom du chromosome et la position sur le chromosome.

Résultats de la validation croisée

L'évaluation de notre méthode est menée par une validation croisée 5-fold sur des ensembles de données humaines et de drosophile. Comme nous avons de grands ensembles de données, nous avons également procédé à une validation croisée 10-fold, et les résultats sont similaires à ceux obtenus avec la validation croisée 5-fold.

Afin d'évaluer la pertinence des noyaux définis, nous avons testé notre méthode en utilisant toutes les combinaison possibles des trois noyaux, y compris l'utilisation d'un seul noyau. Les différents résultats de la validation croisée obtenus sur nos jeux de données d'apprentissage de l'homme et de la drosophile sont donnés dans la Table 5.2. *Km* représente le noyau avec les caractéristiques du k-mer et de l'Uridine, *Kd* représente le noyau de la distance de la séquences par rapport aux centromères et télomères sur le chromosome et *Kn* représente le noyau des k-plus proches voisins. Nous avons comparé notre méthode *piRPred* à l'outil développé par Zhang et ses collaborateurs [215] basé de la méthode k-mer. Afin que ce dernier soit testé dans les mêmes conditions que notre outil, il a été ré-appris sur nos ensembles de données et une validation croisée 5-fold a été effectuée. Les résultats obtenus sont également donnés dans Table 5.2. Les

résultats de classification sont évalués en utilisant les mesures suivantes : la sensibilité (SE) et la sélectivité (PPV) définies dans la Section 3.2.2 et la précision (ACC) définie dans la Section 4.3.5.

	Humain			*Drosophile*		
Méthode	ACC	SE	PPV	ACC	SE	PPV
Km	0.77	**0.80**	0.75	0.67	0.65	0.65
Kd	0.61	0.72	0.59	0.85	0.81	0.87
Kn	0.74	0.66	0.79	0.82	0.81	0.81
Km/Kp	0.77	**0.80**	0.75	0.83	0.78	0.85
Km/Kn	0.79	0.76	0.80	0.82	**0.81**	0.81
Kp/Kn	0.75	0.67	0.80	**0.87**	**0.81**	**0.91**
Km/Kp/Kn	**0.80**	0.77	**0.81**	**0.87**	**0.81**	**0.91**
Zhang *et al.*	0.55	0.27	0.60	0.69	0.44	0.84

TABLE 5.2 – Résultats de la validation croisée obtenus par notre méthode (utilisant différentes combinaisons de noyaux) par la méthode de Zhang *et al.* sur les ensembles de données des espèces *Humain* et *Drosophile. La meilleure valeur pour chaque mesure est donnée en gras.*

Comme le montre la Table 5.2, les résultats de piRPred quelque soit la mesure considérée sont autour de 0,8 aussi bien pour l'homme que pour la drosophile. Nos résultats sont nettement meilleurs que ceux proposés par la méthode de Zhang *et al.* En effet, cet outil échoue sur nos jeux de données d'apprentissages, en particulier sur les données humaines. La précision est proche de 0,5, la valeur d'une classification aléatoire. En outre, la sensibilité est inférieure à 0,5 à la fois chez l'homme et chez la drosophile, ce qui signifie qu'il ne parvient pas à identifier les piARNs positifs. Fait intéressant, en utilisant seulement le noyau *Km*, on a de meilleurs résultats que la méthode de Zhang *et al.*, ce qui confirme les performances supérieures de notre classifieur SVM non linéaire par rapport à la méthode de classification linéaire proposée par Zhang *et al.*.

Comme prévu, les résultats sont légèrement différents entre les séquences humaines et celles de la drosophile, reflétant les différences entre espèces dans les caractéristiques considérées. C'est par exemple le cas du noyau *Kd*, qui donne un meilleur résultat chez la drosophile que chez l'humain. Cette caractéristique mesure la distance d'une séquence donnée par rapport aux régions télomériques/centromérique, et cette tendance a été observée chez la drosophile [19], mais à notre connaissance pas (encore ?) confirmée chez l'homme. Il n'était donc pas sûr que l'application de ce noyau chez l'humain soit bénéfique. Les résultats obtenus chez la drosophile ont démontré des résultats positifs (valeurs supérieures à 0,8), confirmant ainsi son utilité chez la drosophile. Des résultats légèrement positifs ont également été obtenus chez l'homme (valeurs supérieures à 0,5), ce qui suggère une certaine proximité (d'un point de vue statistique) des piARNs par rapport aux régions télomériques et centromériques chez l'humain, mais à un niveau plus faible que chez la drosophile. Inversement, le noyau *Km* donne de meilleurs résultats chez l'homme que chez la drosophile, suggérant une meilleure pertinence des caractéristiques correspondantes, à savoir les fréquences différentielles de certains motifs k-mer et l'apparition d'une Uridine à la première position. Ces résultats correspondent, d'un part, à l'étude que nous avons fait sur la caractéristique de l'Uridine, qui montre que le pourcentage de séquences contenant une Uridine à la première position est plus élevé chez l'homme que chez la drosophile (respectivement 79,68% et 65,93 %), et d'un autre côté, aux résultats publiés dans [215], où il est démontré une meilleure performance de la méthode k-mer proposée sur l'homme que sur la drosophile. Cependant, de façon surprenante, les résultats obtenus par la méthode de Zhang *et al.*, lorsque nous refaisons l'apprentissage sur nos données, donne des résultats complètement inversés. C'est probablement parce que notre ensemble de données d'apprentissage chez la drosophile (composé de 1 300 séquences) est plus grand que celui utilisé par Zhang et ses collaborateurs (composé de 987 séquences), tandis que notre ensemble de données d'apprentissage humain (composé de 7 500 séquences) est plus petit que celui utilisé par Zhang et ses collaborateurs (composé de 32 046 séquences).

Enfin, les résultats obtenus avec la combinaison des trois noyaux sont meilleurs que ceux obtenus par chacun des noyaux utilisés séparément, à la fois chez l'humain et chez la drosophile, montrant une certaine pertinence de leur combinaison.

Sensibilité de la prédiction sur de nouvelles séquences

Nous avons évalué notre algorithme sur 200 séquences de piARNs de l'humain et de la drosophile distinctes de celles utilisées dans l'étape d'apprentissage. Là encore, cela a été fait en comparaison à la méthode de Zhang *et al.*, dont nous avons testé à la fois l'outil disponible sur leur serveur web [48], ainsi que le modèle obtenu après un ré-apprentissage sur nos jeux de données. Les résultats obtenus sont donnés dans Table 5.3.

	Humain		*Drosophile*	
Méthode	TP	SE	TP	SE
piRPred	**159**	**0.80**	**161**	**0.81**
Zhang *et al.* avec modèle sur server web	153	0.77	158	0.79
Zhang *et al.* avec modèle réapprit	55	0.28	61	0.31

TABLE 5.3 – La performance prédictive de(*piRPred*) sur les séquences de l'*Humain* et de la *Drosophile*, en comparaison avec la méthode de Zhang *et al.*

Les résultats de prédiction obtenus par *piRPred* et par la version ré-apprise de la méthode de Zhang *et al.* sont semblables à ceux obtenus dans la validation croisée. Ils montrent une nette meilleure performance de notre algorithme. La méthode de Zhang *et al.* échoue complètement à prédire les piARNs à partir des séquences considérées. Toutefois, lorsqu'on utilise la version du serveur web, les résultats sont meilleurs. La raison pourrait être parce que Zhang et ses collaborateurs ont appris leur méthode sur un très grand ensemble de données (173 090 séquences dont 32 046 séquences humaines et 987 séquences de drosophile), qui incluent probablement les séquences données en test.

Robustesse de la valeur de k dans le noyau des k-plus proches voisins

La valeur de k dans le noyau des k-plus proches voisins (*Kn*) représente le nombre de piARNs dans un cluster sur un brin de chromosome. Cette valeur est très variable, comme on peut le voir dans piRNABank. Certains clusters ne contiennent que deux ou trois piARNs sur un brin, et d'autres en contiennent plusieurs centaines [56]. Afin de déterminer la valeur de k, nous avons effectué plusieurs tests et les résultats obtenus avec différentes valeurs de k sont donnés dans Figure 5.5. A l'issue de ces tests, nous avons décidé de choisir la plus petite valeur pour laquelle la précision est maximale. Comme nous pouvons le voir dans Figure 5.5, la précision reste assez stable à partir de la valeur $k = 4$. Nous avons donc choisi cette valeur comme valeur par défaut.

Ce paramètre peut être réglé par l'utilisateur, et d'autres choix peuvent être faits selon les caractéristiques utilisées dans le but d'améliorer la sensibilité ou la spécificité.

5.3.4 Conclusion

Nous avons proposé un nouvel algorithme appelé piRPred pour l'identification des séquences de piARNs, basé sur une approche combinant la fusion de multiples noyaux avec un classifier SVM. Cette approche permet de tenir compte de caractéristiques hétérogènes, chaque noyau implémentant une classe de caractéristiques. Notre méthode est donc modulaire, extensible et adaptative, permettant la prise en compte de nouvelles caractéristiques, et l'utilisation des caractéristiques les plus appropriées selon l'espèce étudiée.

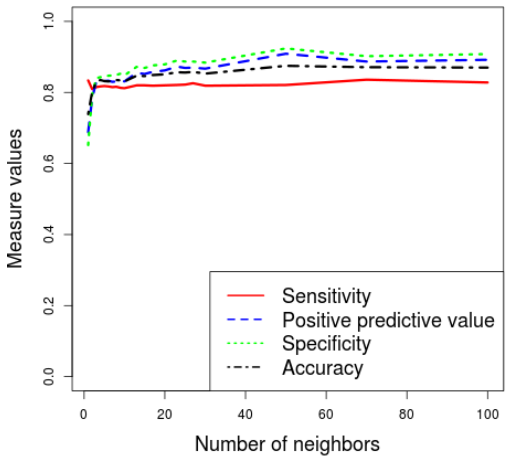

FIGURE 5.5 – Résultats obtenus par le noyau des k-plus proches voisins par différentes valeurs de k sur les données de l'*Humain* (à gauche) et de la *Drosophile* (à droite).

Dans le cas des piRNAs, dont la maturité des connaissances est loin d'avoir atteint celle des miARNs par exemple, il est en effet appréciable de pouvoir prendre en compte les nouvelles études et découvertes sur ces ARNs au fur et à mesure qu'elles sont publiées.

L'un de nos futurs travaux est de rechercher dans la littérature d'autres éventuelles caractéristiques des piARNs, permettant de définir de nouveaux noyaux et ainsi d'améliorer les résultats de prédiction. Un exemple de caractéristique que nous souhaitons explorer est une observation faite par Betel *et al.* sur la souris, qui suggère que 25% des clusters de piARNs sont délimités par des répétitions inversées de longueur variable [13].

Enfin, dans la version actuelle de *piRPred*, l'entrée de l'algorithme est un ensemble de séquences, avec pour chaque séquence sa position sur le génome, incluant des informations sur le brin ("+" ou "-"), le chromosome et la position sur le chromosome. L'algorithme retourne pour chaque séquence donnée la valeur 1 ou 0, selon qu'elle est prédite comme un piARN ou non. Nous projetons d'étendre l'entrée et la sortie de notre algorithme afin de : (i) pouvoir considérer en entrée des données de deep sequencing, et (ii) de pouvoir retourner des clusters de piARNs.

Chapitre 6

Plateforme logicielle EvryRNA

La bioinformatique est un domaine de recherche appliqué, du fait qu'il s'agit ici de développer des méthodes informatiques pour répondre à des problématiques biologiques. Ces méthodes ne doivent donc pas rester à un niveau théorique mais passer à un stade applicatif, et ce par le développement d'outils associés. Malheureusement, dans la recherche académique en général, les outils développés pendant une thèse, un post-doctorat ou tout simplement un projet de recherche (d'une durée en moyenne de trois ans) sont très rarement maintenus après la fin du projet. L'une des raisons est certainement le développement d'outils restant au stade prototype utilisé seulement par le concepteur, et donc non finie pour une utilisation par d'autres personnes.

En bioinformatique, une chose très positive est qu'une bonne partie des journaux les plus importants du domaine obligent la mise à disposition des reviewers mais aussi des lecteurs des outils auxquels font référence le papier que l'on souhaite soumettre. Et heureusement un bon nombre de ces outils restent pérennes bien après la publication de l'article, et souvent via un serveur web permettant soit de télécharger l'outil, soit de l'utiliser à distance. Nous avons pour notre part eu le souci de développer des outils associés à nos algorithmes qui aient un réel intérêt pour la biologie et donc utilisables par les biologistes et les bioinformaticiens, et ce aussi longtemps que possible.

Un serveur web pour Tfold a été initialement développé par un stagiaire de M1 GBI (Gabriel Chandesris), qui a ensuite été étendu et maintenue par d'autres stagiaires (notamment Médérich Besnard et Mikael Trellet) et par un CDD de niveau M2, Frédéric Merle. Ce serveur web a été ensuite étendu par Médérich Besnard, Mikael Trellet et Sébastien tempel pour devenir la plateforme EvryRNA pouvant accueillir les différents logiciels que nous développons.

La plateforme EvryRNA est à l'adresse web `http://EvryRNA.ibisc.univ-evry.fr/`. On peut y trouver tous les algorithmes présentés dans les chapitres précédents, à savoir : *P-DCfold*, *SSCA*, *Tfold*, *miRNAFold*, *miRBoost*, *ncRNAclassifier* et *piRPred* (voir Figure 6.1).

Il s'agit d'un serveur Web dont le but est une utilisation de nos outils à distance. C'est le cas en l'occurrence de *P-DCfold*, *SSCA*, *Tfold* et *miRNAFold*. A l'heure actuelle, *miRBoost*, *ncRNAclassifier* et *piRPred* ne sont disponibles qu'en téléchargement, mais leur intégration pour une utilisation à distance est également prévue, et ce dans une nouvelle version de la plateforme EvryRNA.

EvryRNA présente également des outils que nous avons développés car nécessaires pour nos travaux, par exemple RNA-SC permettant de comparer une structure secondaire prédite avec une structure secondaire de référence. Cet outil retourne les taux de sélectivité et de sensibilité de la structure prédite par rapport à la structure de référence. Par ailleurs, d'autres outils sont en cours d'intégration. Il y aura un outil (pipeline) permettant d'effectuer une prédiction de miARNs suivie d'une étude différentielle à partir de deux ensembles de séquences. La prédiction se fera grâce à la nouvelle version de miRNAFold (intégrant miRBoost), suivie de ncRNAclassifier (pour éliminer des séquences d'éléments transposables), et l'étude diffé-

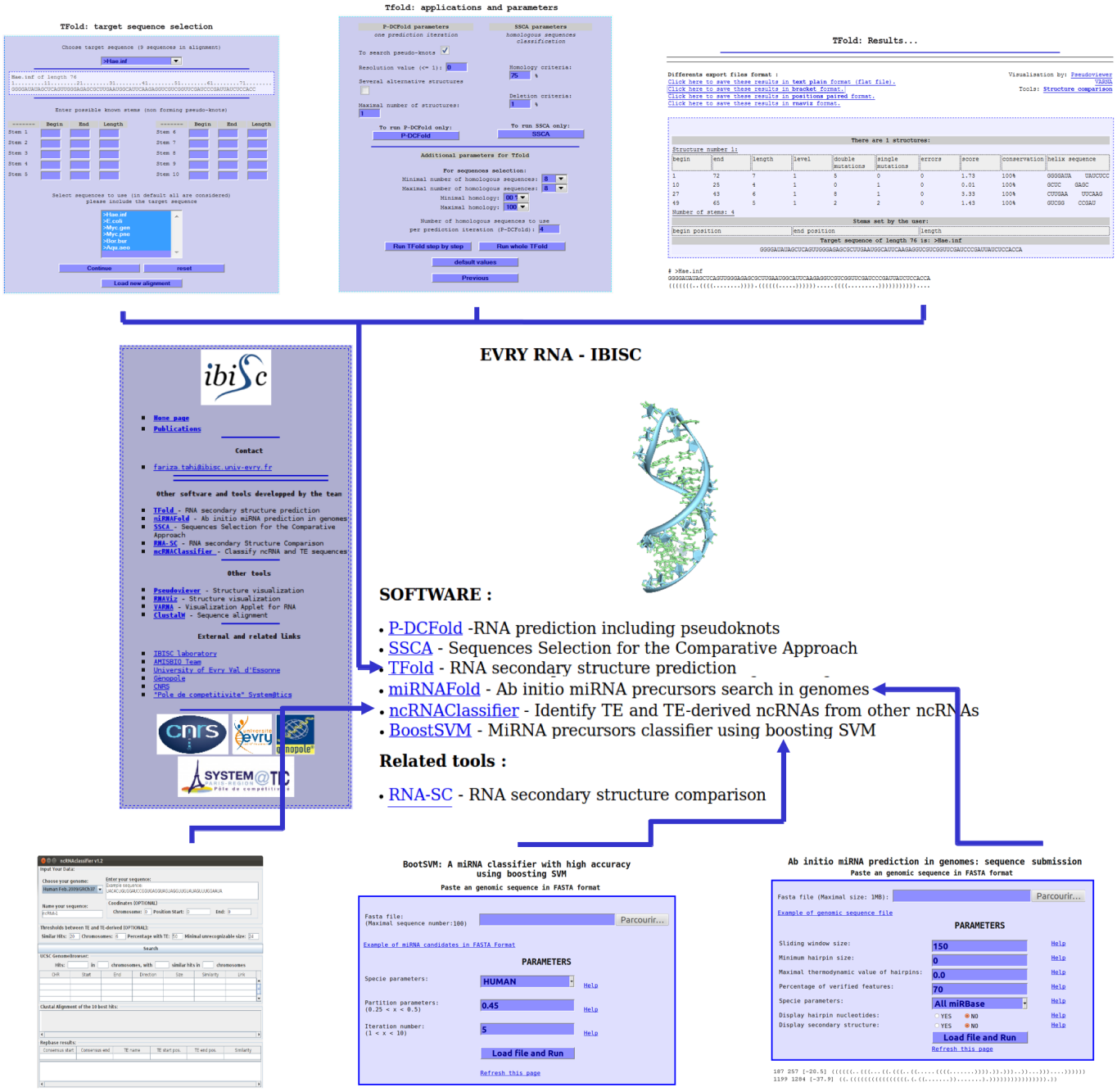

FIGURE 6.1 – La plateforme logicielle EvryRNA

rentielle se fera grâce à l'outil DESeq [6]. Il y aura également un outil permettant de générer des caractéristiques des miARNs d'une espèce donnée, ou éventuellement de plusieurs espèces, à partir des données de miRBase. Ces caractéristiques sont notamment utilisées dans *miRNAFold*.

La plateforme était initialement hébergée chez OVH, puis a été rapatriée à IBISC en 2012. Elle est actuellement maintenue par l'équipe technique du laboratoire. Une plateforme logicielle générique pour tout le laboratoire est en cours de développement par cette équipe dirigée par Laurent Poligny, dans laquelle EvryRNA sera à terme intégrée.

Le logiciel Tfold a été déposé par l'université d'Evry à l'Agence de Protection des Programmes(APP). Un contrat de licence a été signé avec l'Université de Queensland en Australie. Des chercheurs du Mattick

Group de l'Institute for Molecular Bioscience de cette université ont souhaité utiliser Tfold pour l'intégrer dans un pipeline qu'ils développent (comme alternative a RNAalifold dans une analyse pour prédire une structure consensus).

Nous avons par ailleurs fourni le code binaire et/ou le code source de miRNAFold pour plusieurs chercheurs de par le monde qui en ont fait la demande.

Enfin, la plateforme EvryRNA, qui est actuellement signalée essentiellement via nos publications, a été visitée et utilisée plus de 7 000 fois entre octobre 2012 et octobre 2013.

Chapitre 7

Travaux de collaboration en cours

Depuis quelques mois, nous avons débuté deux collaborations avec des biologistes autour de la prédiction d'ARNs non-codants : d'une part avec David Israeli et Laurence Jeanson-Leh du Généthon autour de la recherche de biomarqueurs ARNncs de la Dystrophie Musculaire de Duchenne, et d'autre part avec Abdelhafid Bendahmane et Adnane Boualem de l'URGV autour de la détermination d'ARNncs impliqués dans la différenciation sexuelle chez les plantes. Ces collaborations sont très enrichissantes, et montrent la complexité du monde des ARNs et les défis en bioinformatique que ça soulève.

7.1 Recherche de biomarqueurs ARNncs de la Dystrophie Musculaire de Duchenne

La myopathie de Duchenne, ou dystrophie musculaire de Duchenne (DMD), est une maladie génétique provoquant une dégénérescence progressive de l'ensemble des muscles de l'organisme. Elle a été décrite en 1861 par Guillaume Duchenne. La maladie est liée à une anomalie du gène de la dystrophine, responsable de la production d'une protéine impliquée dans le soutien de la fibre musculaire. Ce gène de 2,5 Mb et 79 exons est le plus grand du génome humain. Il code un acide ribonucléique messager (ARNm) de 14 Kb. La myopathie de Duchenne touche tous les muscles. En effet, en l'absence de dystrophine, les fibres qui composent les muscles squelettiques, lisses et cardiaques se dégradent à chaque contraction et finissent par se détruire. Un mécanisme utilisant des cellules souches musculaires essaye de reconstruire le tissu musculaire endommagé mais la dégénérescence finit toujours par l'emporter. L'espérance de vie avec cette maladie est de l'ordre de 25 à 30 ans à cause d'atteintes respiratoires et cardiaques. Le gène codant la dystrophine est situé sur le chromosome X, cela implique que la maladie touche dans 99,9% des cas des garçons. Les femmes peuvent avoir une mutation dans ce gène mais comme elles possèdent deux chromosomes X, elles ont deux copies du gène. Pour développer la maladie, il faudrait que ces deux gènes soient mutés, ce qui extrêmement rare. Environ 65% des DMD sont dues a des délétions d'une partie du gène de la dystrophine. Cela modifie le cadre de lecture, ce qui entraîne la production d'un ARNm instable et l'absence de la protéine.

La DMD est la plus répandue des myopathies de l'enfant avec 1 garçon sur 3 500 atteint à la naissance. La DMD est l'une des maladies rares étudiées actuellement au Généthon, et plusieurs équipes s'intéressent à la recherche de potentiels biomarqueurs. Les biomarqueurs sont des molécules présentes de manière différentiée dans le sérum ou l'urine chez les sujets sains et les sujets malades, et qui permettent donc de déterminer si un sujet est atteint par la maladie ou pas, et si oui, de donner éventuellement un indice sur l'état d'avancement de la maladie.

Les biomarqueurs concernent essentiellement des protéines, mais de plus en plus d'études ont montré que les miARNs sont aussi de potentiels biomarqueurs, au même titre que les protéines, notamment pour le

cancer [58, 83]. Il s'avère en effet que des miARNs peuvent également être présents dans le sérum et l'urine [131]. Normalement, dans ces milieux, il existe des ribonucléases en quantités importantes qui dégradent les ARNs. Les ARNs qui sont retrouvés doivent donc leur stabilité à une inclusion dans des vésicules ou à une liaison avec des protéines [32]. Ces ARNs sont dits ARNs circulants, et leur détermination a ouvert tout un champ de recherche en biomédical.

Concernant l'étude de la DMD au Généthon, on s'intéresse à déterminer des biomarqueurs aussi bien miARNs que protéines. L'équipe de David Israeli s'intéresse aux miARNs, et travaille sur la détermination de miARNs différentiellement exprimés chez les sujets sains et les sujets malades. La première étape consiste donc à identifier les miARNs présents dans les échantillons de sérum et d'urine dont dispose le Généthon, et cette identification ne peut se faire sans utiliser dans un premier temps des outils bioinformatiques. Nous collaborons avec David Israeli et son équipe sur cette problématique, et co-encadrons actuellement un stage de M2 de Bioinformatique (GBI), Jocelyn Brayet.

Nous avons commencé par appliquer sur les données du Généthon les outils que nous avons déjà développés, à savoir miRNAFold, miRBoost et ncRNAclassifier, afin de prédire de nouveaux miARNs autres que ceux qui sont déjà connus et répertoriés dans la base de données miRBase. Des études différentielles entre les pré-miARNs prédits chez les sujets sains et les pré-miARNs prédits chez les sujets malades ont montré des résultats très intéressants, qui sont en cours d'exploitation.

Par ailleurs, et contre toute attente, d'autres ARNncs, tels que les piARNs, ont été trouvés dans les échantillons de sang et d'urine dont dispose le Généthon. Très vite s'est donc posé le problème d'identification d'éventuels nouveaux piARNs, autres que ceux déjà connus (répertoriés dans la base piRNABank). Nous avons ainsi développé une première version d'un algorithme *ab initio* (appelé piRPred) basé sur de l'apprentissage statistique pour prédire les piARNs (voir Section 5.3). Il est basé sur les SVM et la fusion de multiples noyaux, et les premiers résultats obtenus sur des données d'humain et de drosophile sont très prometteurs. Ce travail n'est qu'à son début, et soulève encore plusieurs challenges : (i) définir d'autres caractéristiques de piARNs et les représenter dans de nouveaux noyaux ; (ii) apprendre automatiquement le poids de chaque noyau et donc de chaque caractéristique pour une meilleure combinaison des différents noyaux ; (iii) tenir compte de grands volumes de données, car en effet les volumes des données positives et négatives de piARNs sont très importants, et ne peuvent être pris en compte par les méthodes classiques d'apprentissage.

7.2 Détermination d'ARNncs impliqués dans la différenciation sexuelle chez les plantes

Le déterminisme sexuel d'une fleur ou d'une plante est un problème central en biologie. Le déterminisme du sexe est le processus qui aboutit à une séparation physique entre les structures produisant les gamètes mâles et femelles dans des fleurs séparées sur la même plante (espèces monoïques) ou sur des plantes individualisées (espèces dioïques). Comprendre ce processus a des implications agronomiques non négligeables. En effet, les productions de plantes hybrides nécessitent par exemple une pollinisation contrôlée, impliquant la production de fleurs mâles stériles (au moyen de la stérilité mâle cytoplasmique). Les systèmes de détermination du sexe sont donc des alternatives intéressantes à la stérilité mâle cytoplasmique.

La grande majorité des végétaux sont hermaphrodites et ont donc des fleurs qui comportent à la fois des organes mâles et femelles. Certaines espèces présentent néanmoins des fleurs de sexes séparés, soit sur la même plante (monoécie), soit sur des plantes différentes (dioécie). La famille des Cucurbitacées (qui comprend la pastèque, le melon, le concombre, la courgette, . . .) est un modèle intéressant pour l'étude de la différenciation sexuelle. Elle présente une grande variabilité de types sexuels. Chez le melon (Cucumis melo), la plupart des variétés traditionnelles cultivées dans le monde sont soit monoïques (présence de fleurs femelles et de fleurs mâles sur la plante), soit andromonoïques (présence de fleurs hermaphrodites et de fleurs mâles sur la plante).

L'équipe de Abdelhafid Bendahmane de l'URGV s'intéresse au déterminisme sexuel chez les cucurbitacées, et est actuellement coordinateur d'une ERC Advanced Grant sur ce sujet, l'ERC SEXYPARTH ("Unraveling sex determination and parthenocarpy mechanisms to improve crops") (2013-2018). Elle a déjà mené plusieurs travaux pour la détermination des gènes impliqués dans la différentiation sexuelle chez le melon et le concombre. Elle a déjà identifié plusieurs gènes impliqués dans ce déterminisme [15, 16, 122, 36] et cherche à déterminer tout le réseau de gènes qui régule ce processus. Les résultats obtenus suggèrent en effet l'existance de plusieurs gènes de sexe interagissant pour contrôler le développement des organes mâles, femelles et hermaphrodites au niveau de la fleur et au niveau de la plante. L'implication des ARNncs, et en particulier des microARNs dans le déterminisme sexuel chez les plantes n'avait pas encore été abordée dans l'équipe de A. Bendahmane. Or il semblerait que les microARNs y jouent un rôle très important, comme le montrent deux récents travaux, l'un effectué sur le maïs [29], et l'autre sur le peuplier [173].

Nous collaborons avec Abdelhafid Bendahmane et Adnane Boualem sur cette thématique et co-encadrons la thèse de Guillaume Beaumont et le stage de M1 de Benjamin Istace pour identifier, via une approche bioinformatique intégrée, le réseau de gènes et aussi d'ARNs non-codants, en particulier les miARNs, qui régulent ce processus de différenciation sexuelle.

Nous travaillons actuellement sur l'identification des miARNs présents dans leurs données, et plus précisément ceux différentiellement exprimés dans les trois sexes. L'URGV dispose à ce jour des génomes entiers du melon, concombre et pastèque, ainsi que de données d'EST ("Expressed Sequence Tag") pour chacun des sexes : femelle, mâle et hermaphrodite. Il s'agit donc d'une part de rechercher les miARNs sur les trois génomes, et d'autre part à partir des données d'EST.

Nous disposons ici de très gros volumes de données. En effet, on est face à des génomes faisant chacun entre 200 et 400 millions de pb, et des données d'EST de plusieurs dizaines de millions de séquences par sexe et par espèce. Sur de tels volumes de données nous sommes confrontés à deux problématiques : le grand nombre de miARNs prédits et donc de faux positives et les temps d'exécution.

Les miARNs chez les plantes présentant des caractéristiques différentes de celles des miARNs chez les animaux, nous développons des versions de miRNAFold et de miRBoost plus adaptées aux plantes. En effet, les versions dont nous disposons actuellement prennent en compte des caractéristiques globales à tous les miARNs connus (répertoriés dans miRBase).

En outre, on dispose ici de plusieurs génomes homologues, on doit donc prendre en compte cette richesse de données pour améliorer la sélectivité de nos algorithmes en y intégrant une phase de comparaison dans les génomes, et identifier donc les miARNs conservés dans les trois génomes. Une version intégrée de miRNAFold et miRBoost est également en cours de développement, ce qui permettrait de réduite le temps global du processus d'analyse.

Bibliographie

[1] J. P. Abrahams, M. Van den Berg, E. Van Batenburg, and C. W. A. Pleij. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucl Acids. Res.*, 18:3035–3044, 1990.

[2] S. Agarwal, C. Vaz, A. Bhattacharya, and A. Srinivasan. Prediction of novel precursor miRNAs using a context-sensitive hidden markov model (CSHMM). *BMC Bioinformatics*, 11(Suppl 1):S29, 2010.

[3] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Mach. Learn. ECML 2004*, volume 3201 of *Lect. Notes Comput. Sci.*, pages 39–50. Springer, Berlin Heidelberg, 2004.

[4] S. Altman, L. Kirsebom, and S. Talbot. Recent studies of Ribonuclease P. *FASEB J.*, 7:7–14, 1993.

[5] Charles Ling And, Charles X. Ling, and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Proc. Fourth Int. Conf. Knowl. Discovery Data Min.*, pages 73–79, New York, USA, 1998.

[6] S Anders and W Huber. Differential expression analysis for sequence count data. *Genome Biol.*, 11(10):R106, 2010.

[7] A Aravin, D Gaidatzis, S Pfeffer, M Lagos-Quintana, P Landgraf, N Iovino, P Morris, MJ Brownstein, S Kuramochi-Miyagawa, T Nakano, M Chien, JJ Russo, J Ju, R Sheridan, C Sander, M Zavolan, and T. Tuschl. A novel class of small RNAs bind to MILI protein in mouse testes. *Nature*, 442(7099):203–7, 2006.

[8] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classication: an overview. *Bioinformatics*, 16:412–424, 2000.

[9] D. Bartel. MicroRNAs: genomics, biogenesis, mechanism and function. *Cell*, 116(2):281–197, 2004.

[10] D Bartel. MicroRNAs: genomics, biogenesis, mechanism and function. *Cell*, 116:281–297, 2004.

[11] Rukshan Batuwita and Vasile Palade. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, 25(8):989–995, 2009.

[12] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Mach. Learn.*, 36:105–139, 1999.

[13] D Betel, R Sheridan, DS Marks, and C. Sander. Computational analysis of mouse piRNA sequence and biogenesis. *Cancer Lett.*, 336(1):46–52, 2013.

[14] biomart. http://www.ensembl.org/biomart.

[15] A. Boualem, M. Fergany, R. Fernandez, C Troadec, A Martin, H Morin, MA Sari, F Collin, JM Flowers, M Pitrat, MD Purugganan, C Dogimont, and A. Bendahmane. A conserved mutation in an ethylene biosynthesis enzyme leads to andromonoecy in melons. *Science*, 321(5890):836–8, 2008.

[16] A. Boualem, C Troadec, I Kovalski, MA Sari, R Perl-Treves, and A. Bendahmane. A conserved mutation in an ethylene biosynthesis enzyme leads to andromonoecy in melons. *Plos one*, 4(7):e6144, 2009.

[17] Markus Brameier and Carsten Wiuf. Ab initio identification of human microRNAs based on structure motifs. *BMC Bioinformatics*, 8(1):478, 2007.

[18] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24:123–140, 1996.

[19] J Brennecke, AA. Aravin, A. Stark, M. Dus, M. Kellis, R. Sachidanandam, and GJ. Hannon. Discrete small RNA-generating loci as master regulators of transposon activity in Drosophila. *Cell*, 128(6):1089–103, 2007.

[20] J.W. Brown. The ribonuclease P database. *Nucl. Acids Res.*, 27:314, 1999.

[21] M. Brown and C. Wilson. RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In *Proceedings of the Pac Symp Biocomput.*, pages 109–25, 1996.

[22] MA Carmell, A Girard, HJG van de Kant, D BourcÕhis, TH Bestor, DG de Rooij, and GJ Hannon. MIWI2 is essential for spermatogenesis and repression of transposons in the mouse male germline. *Cell*, 12(4):503–14, 2007.

[23] caRNAc. http://bioinfo.lifl.fr/carnac/carnac.php.

[24] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:1–27, 2011.

[25] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[26] Shi Chen, Jinqiao Wang, Yang Liu, Changsheng Xu, and Hanqing Lu. Fast feature selection and training for adaboost-based concept detection with large scale datasets. In *Proc. Int. Conf. Multimedia*, MM '10, pages 1179–1182, New York, NY, USA, 2010.

[27] X. Chen, SM. He, D. Bu, F. Zhang, Z. Wang, R. Chen, and W. Gao. Flexstem: improving predictions of RNA secondary structures with pseudoknots by reducing the search space. *Bioinformatics*, 24(18):1994–2001, 2008.

[28] Y Chen, F Zhou, G Li, and Y Xu. A recently active miniature inverted-repeat transposable element, chunjie, inserted into an operon without disturbing the operon structure in geobacter uraniireducens rf4. *Genetics*, 179:2291–7, 2008.

[29] G Chuck, R Meeley, E Irish, H Sakai, and S. Hake. The maize tasselseed4 microRNA controls sex determination and meristem cell fate by targeting Tasselseed6/indeterminate spikelet1. *Nat Genet.*, 39(12):1517–21, 2007.

[30] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[31] L Craig, N, R Gragie, M Gellert, and M. Lambowitz, A. *Mobile DNA II. Second Edition*. ASM Press, ISBN: 1555812090, 2002.

[32] E.E. Creemers, A.J. Tijsen, and YM. Pinto. Circulating MicroRNAs : Novel Biomarkers and Extracellular Communicators in Cardiovascular Disease? . *Circ Res.*, 110(3):483–495, 2012.

[33] F. Crick. The origin of the genetic code. *J. Mol. Biol.*, 38(3):367–379, 1968.

[34] CUDA. http://www.nvidia.fr/object/cuda-parallel-computing-fr.html.

[35] W. C. Curtis and J. N. Vournakis. Quantitation of base substitutions in eukaryotic 5S rRNA: Selection for the maintenance of RNA secondary structure. *J. Mol. Evol.*, 20:351–361, 1984.

[36] F Dahmani-Mardas, C Troadec, A. Boualem, S Lévêque, AA Alsadon, AA Aldoss, C Dogimont, and A. Bendahmane. Engineering melon plants with improved fruit shelf life using the TILLING approach. *Plos one*, 5(12):e15776, 2010.

[37] S.C. Darr, J.W. Brown, and N.R. Pace. The variations of ribonuclease P. *Trends Biochem*, 17:178–182, 1990.

[38] W Dawson, K Fujiwara, G Kawai, Y Futamura, and K. Yamamoto. A method for finding optimal RNA secondary structures using a new entropy model (vsfold). *Nucleosides Nucleotides Nucleic Acids*, 25(2):171–89, 2006.

[39] C. Delisi and D.M. Crothers. Prediction of RNA secondary structure. *Proc. Natl. Acad. Sci.*, 68:2682–2685, 1971.

[40] W Deng and H. miwi Lin. miwi, a murine homolog of piwi, encodes a cytoplasmic protein essential for spermatogenesis. *Developmental Cell*, 2:819–30, 2002.

[41] Jiandong Ding, Shuigeng Zhou, and Jihong Guan. MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC Bioinformatics*, 11(Suppl 11):S11, 2010.

[42] CB. Do, CS. Foo, and S. Batzoglou. A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics*, 24(13):68–76, 2008.

[43] J. P. Dumas and J. Ninio. Efficient algorithms for folding and comparing nucleic acid sequences. *Nucleic Acids Res.*, 10(1):197–206, 1982.

[44] T. Elgavish, J. J. Cannone, J. C. Lee, S. C. Harvey, and R. R. Gutell. AA.AG@Helix.Ends: A:A and A:G base-pairs at the ends of 16S and 23S rRNA helices. *J. Mol. Biol.*, 310:735–753, 2001.

[45] S Engelen and F Tahi. An open problem in RNA secondary structure prediction by the comparative approach. In *Proc. Int. Conf. Math. Eng. Tech. Med. Biol. Sci. (METMBS)*, pages 293–299, 2004.

[46] S. Engelen and F. Tahi. Predicting RNA secondary structure by the comparative approach: how to select the homologous sequences. *BMC Bioinf.*, 8:464, 2007.

[47] S. Engelen and F. Tahi. Tfold: efficient in silico prediction of non-coding RNA secondary structures. *Nucleic Acids Res.*, 38(7):2453–66, 2010.

[48] Zhang et al. http://122.228.158.106/pirna/analysis.php.

[49] B. Felden, H. Himeno, A. Muto, J. McCutcheon, J. Atkins, and R. Gesteland. Probing the structure of the Escherichia coli 10Sa RNA (tmRNA). *RNA*, 3:89–103, 1997.

[50] P. Flajolet, P. Kirschenhofer, and R.F. Tichy. Deviations from uniformity in random strings. *Probability Theory and Related Fields*, 80:139–150, 1988.

[51] S.M. Freier, R. Kierzeck, J.A. Jaeger, N. Sugimoto, M.M. Caruthers, T. Neilson, and D.H. Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl. Acad. Sci.*, 83:9373–9377, 1986.

[52] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. Second Eur. Conf. Comput. Learn. Theory*, pages 23–37, London, UK, 1995.

[53] A Fujita, P, B Rhead, S Zweig, A, S Hinrichs, A, D Karolchik, S Cline, M, M Goldman, P Barber, G, H Clawson, and A *et al.* Coelho. The UCSC Genome Browser database: update 2011. *Nucleic Acids Res.*, 39:D876–D882, 2011.

[54] P. P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *Bioinformatics Online*, 5:140, 2004.

[55] D. Gautheret, D. Konings, and R. R. Gutell. G.U base pairing motifs in ribosomal RNA. *RNA*, 1:807–814, 1995.

[56] A Girard, R Sachidanandam, GJ Hannon, and MA Carmell. A germline-specific class of small RNAs binds mammalian Piwi proteins. *Nature*, 442(7099):199–202, 2006.

[57] T. C. Gluick and D. E. Draper. Thermodynamics of folding a pseudoknotted mRNA fragment. *Journal of Molecular Biology.*, 241:246–262, 1994.

[58] NJ Gooderham and C Koufaris. Using microRNA profiles to predict and evaluate hepatic carcinogenic potential. *Toxicol Lett.*, In Press., 2014.

[59] J. Gorodkin, L.J. Heyer, and G.D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, 25:3724–3732, 1997.

[60] J. Gorodkin, B. Knudsen, C. Zwieb, and T. Samuelsson. SRPDB (Signal Recognition Particle Database. *Nucl. Acids Res.*, 29:169–170, 2001.

[61] L. Grate. Automatic RNA secondary structure determination with stochastic context-free grammars. In *Proc. Third International Conference on Intelligent Systems for Molecular Biology*, pages 136–144, Cambridge, England, Jul 1995.

[62] S. Griffiths-Jones, H.K. Saini, S. van Dongen, and Enright A.J. miRBase: tools for microRNA genomics. *Nucleic Acids Res.*, 36(Database issue):D154–D158, 2008.

[63] A. Grundhoff, C.S. Sullivan, and D. Ganem. A combined computational and microarray-based approach identifies novel microRNAs encoded by human gamma-herpesviruses. *RNA*, 12(5):733–750, 2006.

[64] GtRNAdb. http://lowelab.ucsc.edu/gtrnadb/.

[65] D. G. Guan, J. Y. Liao, Z. H. Qu, Y. Zhang, and L. H. Qu. mirExplorer: detecting microRNAs from genome and next generation sequencing data using the AdaBoost method with transition probability matrix and combined features. *RNA Biol.*, 8(5):922–934, 2011.

[66] Adam Gudys, Michal Szczesniak, Marek Sikora, and Izabela Makalowska. HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC Bioinformatics*, 14(1):83, 2013.

[67] R. Gutell, B. Weiser, C. R. Woese, and H. F. Noller. Comparative anatomy of 16-S-like ribosomal RNA. *Progress in Nucleic Acid Research and Molecular Biology*, 32:155–216, 1985.

[68] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[69] K. Han and H. J. Kim. Prediction of common folding structures of homologous RNAs. *Nucleic Acids Research*, 21(5):1251–1257, 1993.

[70] K. Han and H.J. Kim. Prediction of common folding structures of homologous RNAs. *Nucl. Acid Res.*, 21:1251–1257, 1993.

[71] A. O. Harmanci and G. Sharmaand D. H. Mathews. PARTS: Probabilistic alignment for RNA joint secondary structure prediction. *Nucleic Acids research*, 36(7):2406–2417, 2008.

[72] AO. Harmanci, G. Sharma, and DH. Mathews. Efficient pairwise RNA structure prediction using probabilistic alignment constraints in dynalign. *BMC Bioinformatics*, 8:130, 2007.

[73] C. Haslinger. Prediction algorithms for restricted RNA pseudoknots. *PhD Thesis, Universitat Wien*, 2001.

[74] C. Haslinger and P. F. Stadler. RNA structures with pseudo-knots: Graph-theoretical, combinatorial and statistical properties. *Bul. Math. Biol.*, 61:437–467, 1999.

[75] L. He and G. Hannon. microRNAs: small RNAs with a big role in gene regulation. *Nat. Rev. Genet.*, 5(7):522–531, 2004.

[76] L. He, J. M. Thomson, M. T. Hemann, E. Hernando-Monge, D. Mu, S. Goodson, S. Powers, C. Cordon-Cardo, S. W. Lowe, G. J. Hannon, and S. M. Hammond. A microRNA polycistron as a potential human oncogene. *Nature*, 435:828–833, Jun 2005.

[77] Jana Hertel and Peter F. Stadler. Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, 22(14):e197–e202, 2006.

[78] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie / Chemical Monthly*, 125(2):167–188, 1994.

[79] I.L. Hofacker, M. Fekete, C. Flamm, M.A. Huynen, S. Rauscher, P.E. Stolorz, and P.F. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucl. Acids Res.*, 26:3825–3836, 1998.

[80] I.L. Hofacker, W. Fontana, P.F. Stadler, L.S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, 125:167–188, 1994.

[81] K. Horimoto, J. Otsuka, and T. Kunisawa. Rapid evolutionary repair of base mispairings in stem regions of eukaryotic 5S rRNA. *Protein Seq. Data Anal.*, 2:93–99, 1989.

[82] Ting-Hua Huang, Bin Fan, Max Rothschild, Zhi-Liang Hu, Kui Li, and Shu-Hong Zhao. MiRFinder: an improved approach and software implementation for genome-wide fast microRNA precursor scans. *BMC Bioinformatics*, 8(1):341, 2007.

[83] CH Hung, YC Chiu, CH Chen, and TH Hu. MicroRNAs in Hepatocellular Carcinoma: Carcinogenesis, Progression, and Therapeutic Target. *Biomed Res Int.*, 2014(486407), 2014.

[84] H. Jabbari, A. Condon, and S. Zhao. Novel and efficient RNA secondary structure prediction using hierarchical folding. *J Comput Biol*, 15(2):139–63, 2008.

[85] B. D. James, G. J. Olsen, and N. R. Pace. Phylogenetic comparative analysis of RNA secondary structure. *Methods in Enzymology*, 180:227–239, 1989.

[86] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. Int. Conf. Artif. Intell.*, pages 111–117, Las Vegas, Nevada, 2000.

[87] P. Jiang, H. Wu, W. Wang, W. Ma, X. Sun, and Z. Lu. MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.*, 35:W339–344, Jul 2007.

[88] Qinghua Jiang, Yadong Wang, Yangyang Hao, Liran Juan, Mingxiang Teng, Xinjun Zhang, Meimei Li, Guohua Wang, and Yunlong Liu. miR2Disease: a manually curated database for microRNA deregulation in human disease. *Nucleic Acids Res.*, 37(Suppl 1):D98–D104, 2009.

[89] I. Jung, J.C. Park, and S.. Kim. piClust: A density based piRNA clustering algorithm. *Computational Biology and Chemistry*, 2014.

[90] J Jurka, V Kapitonov, V, A Pavlicek, P Klonowski, O Kohany, and J Walichiewicz. Repbase update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research*, 110:462–467, 2005.

[91] J Jurka, P Klonowski, V Dagman, and P Pelton. CENSOR - a program for identification and elimination of repetitive elements from DNA sequences. *Comput Chem.*, 20:119–21, 1996.

[92] P Kapranov, T Willingham, A, and R. Gingeras, T. Genome-wide transcription and the implications for genomic organization. *Nat. Rev. Genet.*, 8:413–23, 2007.

[93] H Kawagoe-Takaki, N Nameki, M Kajikawa, and N Okada. Probing the secondary structure of salmon Smal SINE RNA. *Gene*, 365:67–73, 2006.

[94] J Ken, W. BLAT–The BLAST-Like Alignment Tool. *Genome Res.*, 4:656–664, 2002.

[95] SH Kim, JL Sussman, FL Suddath, GJ Quigley, A McPherson, AH Wang, NC Seeman, and A. Rich. The general structure of transfer RNA molecules. *PNAS*, 71(12):4970–4974, 1974.

[96] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, 31(13):3423–3428, 2003.

[97] B. Knudsen, J., and Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research.*, 31(13):3423–3428, 2003.

[98] Ana Kozomara and Sam Griffiths-Jones. miRBase: integrating microRNA annotation and deep-sequencing data. *Nucleic Acids Res.*, 39(suppl 1):D152–D157, 2011.

[99] H Kuang, C Padmanabhan, F Li, A Kamei, B Bhaskar, P, S Ouyang, J Jiang, R Buell, C, and B Baker. Identification of miniature inverted-repeat transposable elements (MITEs) and biogenesis of their siRNAs in the Solanaceae: New functional implications for MITEs. *Genome Res.*, 19:42–56, 2009.

[100] S Kuramochi-Miyagawa, T Kimura, TW Ijiri, T Isobe, N Asada, Y Fujita, M Ikawa, N Iwai, M Okabe, W Deng, H Lin, Y Matsuda, and T. Nakano. Mili, a mammalian member of piwi family gene, is essential for spermatogenesis. *Developmental*, 131(4):839–49, 2004.

[101] Eric Lai, Pavel Tomancak, Robert Williams, and Gerald Rubin. Computational identification of Drosophila microRNA genes. *Genome Biol.*, 4(7):R42, 2003.

[102] S. S. Lakshmi and S. Agrawal. piRNABank: a web resource on classified and clustered Piwi-interacting RNAs. *Nucleic Acids Res.*, 36(Database issue):D173–D177, 2008.

[103] P Landgraf, M Rusu, R Sheridan, A Sewer, N Iovino, A Aravin, S Pfefferand, A Rice, AO Kamphorst, M Landthaler, and *et al.* A Mammalian microRNA Expression Atlas Based on Small RNA Library Sequencing. *Cell*, 129:1401–14, 2007.

[104] D Langenberger, S Bartschat, J Hertel, S Hoffmann, H Tafer, and F Stadler, P. microrna or not microrna? In N de Sousa, GP Telles, and MJ Palakal, editors, *Advances in Bioinformatics and Computational Biology*, pages 1–9. Springer, Heidelberg, 6th bsb edition, 2011.

[105] A Larkin, M, G Blackshields, P Brown, N, R Chenna, A McGettigan, P, H McWilliam, F Valentin, M Wallace, I, A Wilm, R Lopez, D Thompson, J, J Gibson, T, and G Higgins, D. ClustalW and ClustalX version 2.0. *Bioinformatics*, 23:2947–8, 2007.

[106] N. Larsen, G. J. Olsen, B. L. Maidak, M. J. McCaughey, R. Overbeek, T. J. Macke, T. L. Marsh, and C. R. Woese. The ribosomal database project. *NAR*, 21(1):3021–3023, 1993.

[107] N. Larsen, G.J. Olsen, B.L. Maidak, M.J. McCaughey, R. Overbeek, T.J. Macke, T.L. Marsh, and C.R. Woese. The ribosomal database project. *Nucl. Acid Res.*, 21:3021–3023, 1993.

[108] NC Lau, AG Seto, J Kim, S Kuramochi-Miyagawa, T Nakano, DP Bartel, and RE. Kingston. Characterization of the piRNA complex from rat testes. *Science*, 313(5785):363–07, 2006.

[109] Y. Lee, M. Kim, J. Han, K. Yeom, S. Lee, S. Baek, and V. Kim. microRNA genes are transcribed by RNA polymerase II. *EMBO J.*, 23(20):4051–4060, 2004.

[110] Matthieu Legendre, André Lambert, and Daniel Gautheret. Profile-based detection of microRNA precursors in animal genomes. *Bioinformatics*, 21(7):841–845, 2005.

[111] N.B. Leontis and E. Westhof. A common motif organizes the structure of multi-helix loops in 16S and 23S ribosomal RNAs. *J. Mol. Biol.*, 283:571–583, 1998.

[112] S. Lertampaiporn, C. Thammarongtham, C. Nukoolkit, B. Kaewkamnerdpong, and M. Ruengjitchatchawalya. Heterogeneous ensemble approach with discriminative features and modified-SMOTEbagging for pre-miRNA classification. *Nucleic Acids Res.*, 41(1):e21, Jan 2013.

[113] A LeThomas, KF Toth, and AA Aravin. To be or not to be a piRNA: genomic origin and processing of piRNAs. *Genome Biol.*, 15(204):47–58, 2014.

[114] A Levy, N Sela, and G Ast. Transpogene and microtranspogene: transposed elements influence on the transcriptome of seven vertebrates and invertebrates. *Nucleic Acids Res.*, 36:D47–52, 2007.

[115] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, 2004.

[116] Xuchun Li, Lei Wang, and Eric Sung. AdaBoost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.*, 21:785–795, 2008.

[117] Yaoyong Li and John Shawe-Taylor. The SVM with uneven margins and Chinese document categorization. In *Proc 17th Pac. Asia Conf. Lang. Inf. Comput.*, pages 216–227, Singapore, 2003.

[118] L. P. Lim, N. C. Lau, E. G. Weinstein, A. Abdelhakim, S. Yekta, M. W. Rhoades, C. B. Burge, and D. P. Bartel. The microRNAs of Caenorhabditis elegans. *Genes Dev.*, 17:991–1008, Apr 2003.

[119] R.B. Lyngso and C.N.S. Pedersen. Pseudoknots in RNA secondary structures. In *Proc. RECOMB*, pages 201–209, 2000.

[120] O. C. Maes, H. M. Chertkow, E. Wang, and H. M. Schipper. MicroRNA: Implications for Alzheimer Disease and other Human CNS Disorders. *Curr. Genomics*, 10:154–168, May 2009.

[121] R. Mans, C. Pleij, and L. Bosch. Transfer RNA-like structures: Structure, function and evolutionary significance. *Eur. J. Biochem.*, 201(2):303–324, 1991.

[122] A Martin, C Troadec, A Boualem, M Rajab, R Fernandez, H Morin, M Pitrat, C Dogimont, and A. Bendahmane. A transposon-induced epigenetic change leads to sex determination in melon. *Nature*, 461(7267):1135–8, 2009.

[123] H. M. Martinez. An RNA secondary structure workbench. *Nucleic Acids Res.*, 16(5):1789–1798, 1988.

[124] A. Mathelier and A. Carbone. MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, 26:2226–2234, Sep 2010.

[125] D.H. Mathews and D.H. Turner. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.*, 317:191–203, 2002.

[126] D.H. Mathews, D.H. Turner, and M. Zuker. RNA secondary structure prediction. *Current Protocols in Nucleic Acid Chemistry*, 11:1–10, 2000.

[127] D.H. Matthews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, 1999.

[128] Y. Mei, D. Clark, and L. Mao. Novel dimensions of piRNAs in cancer. *Cancer Lett.*, 336(1):46–52, 2013.

[129] Mfold. http://mfold.bioinfo.rpi.edu/cgi-bin/rna-form1.cgi.

[130] miRBase. http://www.mirbase.org.

[131] P.S. Mitchell and R.K. Parkin. Circulating microRNAs as stable blood-based markers for cancer detection. *PNA USA*, 105(30):10513Ð10518, 2008.

[132] K Miyoshi, T Miyoshi, and H Siomi. Many ways to generate microRNA-like small RNAs: non-canonical pathways for microRNA production. *Mol. Genet. Genomics*, 284:95–103, 2010.

[133] Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proc. 16th Int. Conf. Mach. Learn.*, pages 268–277, San Francisco, CA, USA, 1999.

[134] M. Mraz, S. Pospisilova, K. Malinova, I. Slapak, and J. Mayer. MicroRNAs in chronic lymphocytic leukemia pathogenesis and disease subtypes. *Leuk Lymphoma*, 50:506–509, Mar 2009.

[135] J. W. Nam, K. R. Shin, J. Han, Y. Lee, V. N. Kim, and B. T. Zhang. Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res.*, 33:3570–3581, 2005.

[136] NCBI. http://www.ncbi.nlm.nih.gov/genome.

[137] K. L. Ng and S. K. Mishra. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, 23:1321–1330, Jun 2007.

[138] P Nissen, J Hansen, N Ban, PB Moore, and TA; Steitz. The structural basis of ribosome activity in peptide bond synthesis. *Sci.*, 289(5481):920–930, 2000.

[139] H. F. Noller. Structure of ribosomal RNA. *Annu. Rev. Biochem.*, 53:119–162, 1984.

[140] H. F. Noller and C. R. Woese. Secondary structure of 16S ribosomal RNA. *Science*, 212:403–411, 1981.

[141] M Nozawa, S Miura, and M Nei. Origins and evolution of microRNA genes in Drosophila species. *Genome Biol. Evol.*, 12:180–09, 2010.

[142] R. Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single-strand RNA. *Proc. Natl. Acad. Sci.*, 77:6309–6313, 1980.

[143] R. Nussinov, G. Pieczenik, J.R. Griggs, and D.J. Kleitman. Algorithm for loop matchings. *SIAM J. Appl. Math.*, 35:68–82, 1978.

[144] K Okamura, W Chung, and C E Lai. The long and short of inverted repeat genes in animals: microRNAs, mirtrons and hairpin RNAs. *Cell Cycle*, 7(18):2840–2845, 2008.

[145] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.*, 11:169–198, 1999.

[146] T. Pavlidis, R. Pace, and D. Smith. Ribonuclease P: function and variation. *J. Biol. Chem.*, 265:3587–3590, 1990.

[147] JC. Peng and H. Lin. Beyond transposons: the epigenetic and somatic functions of the Piwi-piRNA mechanism. *Current Opinion in Cell Biology*, 25:190–4, 2013.

[148] O. Perriquet, H. Touzet, and M. Dauchet. Finding the common structure shared by two homologous RNAs. *Bioinformatics*, 19:108–116, 2003.

[149] Pfold. http://www.daimi.au.dk/ compbio/rnafold/.

[150] J Piriyapongsa and K Jordan, I. A Family of Human MicroRNA Genes from Miniature Inverted-Repeat Transposable Elements. *PLoS ONE*, 2:e203, 2007.

[151] J Piriyapongsa and K Jordan, I. Dual coding of siRNAs and miRNAs by plant transposable element. *RNA*, 14:814–21, 2008.

[152] J Piriyapongsa, L Marino-Ramirez, and K Jordan, I. Origin and Evolution of Human microRNAs From Transposable Elements. *Genetics*, 176:1323–1337, 2007.

[153] piRNABank. http://pirnabank.ibab.ac.in/index.shtml.

[154] pknotsRG. http://bibiserv.techfak.uni-bielefeld.de/pknotsrg/.

[155] U Poolsap, Y Kato, and T Akutsu. Prediction of RNA secondary structure with pseudoknots using integer programming. *BMC Bioinformatics*, 10, 2009.

[156] Pedro Rangel, Fernando Lozano, and Elkin Garcia. Boosting of support vector machines with application to editing. In *Proc. Fourth Int. Conf. Mach. Learn. Appl.*, pages 374–382, Washington DC, USA, 2005.

[157] D.B. Redpath and K. Lebart. Boosting feature selection. In Sameer Singh, Maneesha Singh, Chid Apte, and Petra Perner, editors, *Pattern Recognit. Data Mining*, volume 3686 of *Lect. Notes Comput. Sci.*, pages 305–314. Springer, Berlin Heidelberg, 2005.

[158] J Reeder, P Steffen, and R Giegerich. pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows. *Nucleic Acids Res. (Web Server issue)*, 35, 2007.

[159] E. Rivas and S.Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, 285:2053–2068, 1999.

[160] RNAalifold. http://rna.tbi.univie.ac.at/cgi-bin/rnaalifold.cgi.

[161] D. Rosenkranz and H. Zischler. proTRAC–a software for probabilistic piRNA cluster detection, visualization and analysis. *BMC Bioinformatics*, 13(5), 2012.

[162] RJ. Ross, MM. Weiner, and H. Lin. PIWI proteins and PIWI-interacting RNAs in the soma. *Nature*, 505(7483):353–9, 2014.

[163] F. Rousset, M. PÈlandakis, and M. Solignac. Evolution of compensatory substitutions through G.U intermediate state in drosophila rRNA. *Proc. Natl. Acad. Sci.*, 88:10032–10036, 1991.

[164] J. Ruan, G. Stormo, and W. Zhang. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics.*, 20:58–66, 2004.

[165] D. Sankoff. Simultaneous solution of the RNA folding, aligment and protosequence problems. *SIAM J. Appl. Math.*, 45:810–825, 1985.

[166] Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5:197–227, 1990.

[167] P. Schimmel. RNA pseudoknots that interact with components of the translation apparatus. *Cell I*, 58(1):9–12, 1989.

[168] A Sewer, N Paul, P Landgraf, A Aravin, S Pfeffer, J Brownstein, M, T Tuschl, E van Nimwegen, and M Zavolan. Identification of clustered microRNAs using an ab initio prediction method. *BMC Bioinformatics*, 6(1):267, 2005.

[169] B. A. Shapiro and J. Navetta. A massively parallel genetic algorithm for RNA secondary structure prediction. *The Journal of Supercomputing*, 8:195–207, 1994.

[170] B. A. Shapiro and J. C. Wu. Predicting RNA h-type pseudoknots with the massively parallel genetic algorithm. *Comput Appl Biosci*, 13(4):459–471, 1997.

[171] R Smalheiser, N and I Torvik, V. Mammalian microRNAs derived from genomic repeats. *Trends Genet.*, 21:322–326, 2005.

[172] R Smalheiser, N and I Torvik, V. Alu elements within human mRNAs are probable microRNA targets. *Trends Genet.*, 22:322–6, 2006.

[173] Y Song, K Ma, D Ci, Z Zhang, and D. Zhang. Sexual dimorphism floral microRNA profiling and target gene expression in andromonoecious poplar (Populus tomentosa). *PLoS One.*, 8(5):e62681, 2013.

[174] J Sperschneidern and A Datta. KnotSeeker: heuristic pseudoknot detection in long RNA sequences. *RNA*, 14(4):630–40, 2008.

[175] D Suntera, J, P Patela, S, A Skiltona, R, N Githakaa, P Knowlesb, D, A Scolesb, G, V Nened, E de Villiers, and P Bishopa, R. A novel sine family occurs frequently in both genomic dna and transcribed sequences in ixodid ticks of the arthropod sub-phylum chelicerata. *Gene*, 415:13–22, 2008.

[176] M. Szymanski, M. Z. Barciszewska, V. A. Erdmann, and J. Barciszewski. 5S ribosomal RNA database. *Nucl. Acids Res.*, 30:176–178, 2002.

[177] F Tahi, S Engelen, and M Régnier. A fast algorithm for RNA secondary structure prediction including pseudoknots. In *Proc. IEEE Symp. BioInf. BioEng. (BIBE)*, pages 11–17, 2003.

[178] F. Tahi, S. Engelen, and M. Régnier. P-DCFold or How to predict all kinds of pseudoknots in RNA secondary structures. *Int. J. Artif. Intell. Tools*, 14(5):703–716, 2005.

[179] F. Tahi, M. Gouy, and M. Regnier. Automatic RNA secondary structure prediction with a comparative approach. *Comput. Chem.*, 26(5):521–530, 2002.

[180] F Tahi, V.D. Tran, S Tempel, and E Mahé. *Patterns for parallel programming on GPU's*, chapter Bioinformatics of non-coding RNAs and GPUs. A case study: prediction at large scale of microRNAs in genomes. F. Magoulès, saxe coburg publications edition, 2014.

[181] C. K. Tand and D. E. Draper. An unusal mRNA pseudoknot structure is recognized by a protein translation repressor. *Cell.*, 57:531–536, 1989.

[182] C. K. Tand and D. E. Draper. Evidence for allosteric coupling between the ribosome and repressor binding sites for a translationally regulated mRNA. *Biochemistry.*, 29:4434–4439, 1990.

[183] S. Tempel, N. Pollet, and F. Tahi. ncRNAclassifier: a tool for the detection of transposable element sequences in RNA hairpins and their classification. *BMC Bioinformatics*, 13(246), 2012.

[184] S. Tempel and F. Tahi. An automatic method for identifying TE-derived miRNAs. In *Proc. JOBIM*, pages 245–252, 2011.

[185] S. Tempel and F. Tahi. A fast ab-initio method for predicting miRNA precursors in genomes. *Nucleic Acids Res.*, 40(11):e80, 2012.

[186] S. Tempel and F. Tahi. miRNAFold: A fast ab-initio method for searching for miRNA precursors in whole genomes. In *Proc. JOBIM*, pages 39–46, 2012.

[187] Goro Terai, Takashi Komori, Kiyoshi Asai, and Taishin Kin. miRRim: A novel system to find conserved miRNAs with high sensitivity and specificity. *RNA*, 13(12):2081–2090, 2007.

[188] T. Thum, P. Galuppo, C. Wolf, J. Fiedler, S. Kneitz, L. W. van Laake, P. A. Doevendans, C. L. Mummery, J. Borlak, A. Haverich, C. Gross, S. Engelhardt, G. Ertl, and J. Bauersachs. MicroRNAs in the human heart: a clue to fetal gene reprogramming in heart failure. *Circulation*, 116:258–267, Jul 2007.

[189] Kai Ting and Lian Zhu. Boosting support vector machines successfully. In Juan Benediktsson, Josef Kittler, and Fabio Roli, editors, *Mult. Classifier Syst.*, volume 5519 of *Lect. Notes Comput. Sci.*, pages 509–518. Springer, Berlin Heidelberg, 2009.

[190] I.Jr. Tinoco and O.C. Uhlenbeck. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[191] tmRDB. http://www.ag.auburn.edu/mirror/tmrdb/.

[192] V. D. Tran, B. Zerath, S. Tempel, F. Zehraoui, and F. Tahi. BoostSVM: A miRNA classifier with high accuracy using boosting SVM. In *Proc. JOBIM*, pages 259–266, 2012.

[193] C Tuck, A and D Tollervey. RNA in pieces. *Trends Genetics*, 27:422–432, 2011.

[194] S. Tyagi, C. Vaz, V. Gupta, R. Bhatia, S. Maheshwari, A. Srinivasan, and A. Bhattacharya. CID-miRNA: a web server for prediction of novel miRNA precursors in human genome. *Biochem. Biophys. Res. Commun.*, 372:831–834, Aug 2008.

[195] E. van Rooij. The art of microRNA research. *Circ. Res.*, 108(2):219–234, Jan 2011.

[196] L. Vawter and W. M. Brown. Rates and patterns of base change in the small subunit ribosomal RNA gene. *Genetics.*, 134:597–608, 1993.

[197] O Voinnet. Origin, Biogenesis, and Activity of Plant MicroRNAs. *Cell*, 136:669–687, 2007.

[198] vsfold. http://www.rna.it-chiba.ac.jp/ vsfold/vsfold5/.

[199] Benjamin Wang and Nathalie Japkowicz. Boosting support vector machines for imbalanced data sets. *Knowl. Info. Syst.*, 25:1–20, 2010.

[200] Xiaowo Wang, Jing Zhang, Fei Li, Jin Gu, Tao He, Xuegong Zhang, and Yanda Li. MicroRNA identification based on sequence and structure alignment. *Bioinformatics*, 21(18):3610–3614, 2005.

[201] T Watanabe, A Takeda, T Tsukiyama, K Mise, T Okuno, H Sasaki, N Minami, and H. Imai. Identification and characterization of two novel classes of small RNAs in the mouse germline: retrotransposon-derived siRNAs in oocytes and germline small RNAs in testes. *Genes Dev.*, 20(13):1732–43, 2006.

[202] T Wicker, F Sabot, A Hua-Van, L Bennetzen, J, P Capy, B Chalhoub, A Flavell, P Leroy, M Morgante, O Panaud, E Paux, P San Miguel, and H. Schulman, A. A unified classification system for eukaryotic transposable elements. *Nat. Rev. Genet.*, 8:973–82, 2007.

[203] Jeevani Wickramaratna, Sean Holden, and Bernard Buxton. Performance degradation in boosting. In Josef Kittler and Fabio Roli, editors, *Mult. Classifier Syst.*, volume 2096 of *Lect. Notes Comput. Sci.*, pages 11–21. Springer, Berlin Heidelberg, 2001.

[204] S Will, K Reiche, IL Hofacker, PF Stadler, and Backofen R. Inferring noncoding rna families and classes by means of genome-scale structure-based clustering. *Nucleic Acids research*, 3(4):e65, 2007.

[205] C.R. Woese, S. Winker, and R. R. Gutell. Architecture of ribosomal RNA: constraints on the sequence of tetra-loops. *Proc. Natl. Acad. Sci.*, 87:8467–8471, 1990.

[206] Gang Wu and Edward Y. Chang. Class-boundary alignment for imbalanced dataset learning. In *Proc. Workshop Learn. Imbalanced Data Sets*, pages 49–56, Washington DC, USA, 2003.

[207] Kuo-Ping Wu and Sheng-De Wang. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recognition*, 42(5):710 – 717, 2009.

[208] Yonggan Wu, Bo Wei, Haizhou Liu, Tianxian Li, and Simon Rayner. MiRPara: a SVM-based software tool for prediction of most probable microRNA coding regions in genome scale sequences. *BMC Bioinformatics*, 12(1):107, 2011.

[209] X. Xu, Y. Ji, and GD. Stormo. Discovering cis-regulatory RNA in shewanella genomes by support vector machines. *PLoS Comput Biol*, 5(4):e1000338, 2009.

[210] Yunpen Xu, Xuefeng Zhou, and Weixiong Zhang. MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*, 24(13):i50–i58, 2008.

[211] C. Xue, F. Li, T. He, G. P. Liu, Y. Li, and X. Zhang. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6:310, 2005.

[212] Y Yan, Y Zhang, K Yang, Z Sun, Y Fu, X Chen, and R Fang. Small RNAs from MITE-derived stem-loop precursors regulate abscisic acid signaling and abiotic stress responses in rice. *The Plant Journal*, 65:820–828, 2011.

[213] J. Yang, Z. Luo, X. Fang, J. Wang, and K. Tang. Predicting RNA secondary structures including pseudoknots by covariance with stacking and minimum free energy. *Sheng Wu Gong Cheng Xue Bao*, 24(4):659–64, 2008.

[214] Malik Yousef, Michael Nebozhyn, Hagit Shatkay, Stathis Kanterakis, Louise C. Showe, and Michael K. Showe. Combining multi-species genomic data for microRNA identification using a Naïve Bayes classifier. *Bioinformatics*, 22(11):1325–1334, 2006.

[215] Y. Zhang, X. Wang, and L. Kang. A k-mer scheme to predict piRNAs and characterize locust piRNAs. *Bioinformatics*, 27(6):771–6, 2011.

[216] Y. Zhao and Z. Wang. RNA secondary structure prediction based on support vector machine classification. *Chin. J. Biotechnol.*, 24(7):1140–8, 2008.

[217] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[218] C. Zwieb. The uRNA database. *Nucl. Acids Res.*, 24:76–79, 1996.

[219] C. Zwieb. The uRNA database. *Nucl. Acids Res.*, 24:76–79, 2003.

[220] C. Zwieb, J. Gorodkin, B. Knudsen, J. Burks, and J. Wower. tmRDB (tmRNA database). *Nucl. Acids Res.*, 31:446–447, 2003.