THÈSE

présentée à
l'**U**niversité **L**ouis **P**asteur de Strasbourg
UPR N°9002 du CNRS : **A**rchitecture & **R**éactivité de l'AR**N**
Institut de **B**iologie **M**oléculaire et **C**ellulaire

Pour l'obtention du grade de
Docteur de l'Université Louis Pasteur
Discipline SCIENCES DU VIVANT
Domaine ASPECTS MOLÉCULAIRES ET CELLULAIRES DE LA BIOLOGIE

par

Thomas LUDWIG

Développement d'un environnement bioinformatique dédié à la construction d'architectures d'ARN

Soutenue publiquement le 17 Septembre 2008 devant le jury composé de :

M. **Eric Westhof**, Directeur de thèse
Professeur à l'Université Louis Pasteur de Strasbourg
M. **Marc Delarue**, Rapporteur externe
Directeur de Recherche à l'Institut Pasteur de Paris
M. **Alain Denise**, Rapporteur externe
Professeur à l'Université de Paris-Sud
M. **Jean-Marie Wurtz**, Rapporteur interne
Professeur à l'Université Louis Pasteur de Strasbourg
M. **Fabrice Jossinet**, Examinateur
Maître de Conférences à l'Université Louis Pasteur de Strasbourg

Développement d'un environnement bioinformatique dédié à la construction d'architectures d'ARN

Les découvertes de ces dernières années suggèrent que la complexité et le degré d'évolution d'un organisme sont reliés à l'existence de centaines de milliers d'ARN non codants, de petites et de grandes tailles. Ainsi, de «simple» acteur dans les mécanismes de traduction (via les ARNm, ARNt et ARNr), l'ARN se retrouve être un élément clé dans un nombre toujours croissant de mécanismes régulant les grandes fonctions biologiques et, par voie de conséquence, de mécanismes les perturbant et impliqués dans les phénomènes de cancer, d'infections virales et d'affections neurologiques.
En conséquence, on assiste depuis quelques années au développement de nombreux outils permettant l'identification de nouveaux ARN non codants au sein des génomes séquencés. Ils ont permis de découvrir un nombre très important de nouveaux candidats dans les organismes modèles. La validation expérimentale de ces candidats et la compréhension de leur fonction biologique passe par la détermination de leur architecture tridimensionnelle. Malheureusement, les limites des approches expérimentales actuelles ne permettent pas de répondre assez rapidement à ce besoin de validation.
Dans ce contexte, nous avons décidé de développer un environnement bioinformatique permettant d'optimiser la construction et la compréhension des architectures d'ARN.
La première partie de cette thèse se focalise sur la description de cette infrastructure nommée **P.A.R.A.DIS.E** (***P**latform to **A**nalyze **R**NA **A**nnotations over a **Dis**tributed **E**nvironment*). Son développement peut se subdiviser en trois grands axes :
– un moteur de gestion des annotations d'ARN (structures secondaires et tertiaires, alignements de séquences, ...)
– une couche de communication permettant d'utiliser de manière transparente des algorithmes produisant ces annotations.
– une couche graphique facilitant la visualisation et la manipulation des ces annotations au moyen de représentations adaptées.
La seconde partie est plus particulierment dédiée à la description du module graphique nommé **Assemble** permettant de construire un modèle de structure tridimensionnelle pour une molécule d'ARN dont la structure secondaire est connue. **Assemble** dispose d'un grand nombre d'outils automatiques permettant de réaliser un modèle le plus rapidement possible en rendant possible la génération automatique d'un premier jet de structure tertiaire, l'application de motifs structuraux sur certaines régions du modèle et l'affinement des coordonnées du modèle en accord avec des contraintes structurales. L'utilisateur a également la possibilité d'afficher une carte de densité électronique, servant de repère dans le processus de modélisation.
Enfin, la troisième et dernière partie du manuscrit s'attache à la validation des outils que nous avons développés, en décrivant de façon détaillée leurs applications concrètes à des problématiques biologiques d'actualité autour de l'ARN.

Development of a bioinformatics environment dedicated to the construction of RNA architectures

Recent discoveries suggest that the complexity and degree of evolution of an organism is linked to the existence of hundreds of thousands non coding RNA of various sizes. Thus, from a mere actor in translation mechanisms (through mRNA, tRNA and rRNA), RNA is found to be a key element in an ever increasing number of mechanisms regulating important biological functions, and therefore in mechanisms disturbing them and is involved in cancer, viral infections and neurological pathologies.
In consequence, we witness, since a few year, the development of numerous tools dedicated to the identification of new non coding RNA within the sequenced genomes. Those tools allowed the discovery of a great number of new candidates within model organisms. Experimental validation of these candidates and understanding of their biological function rely on the determination of their three-dimensional architectures. Unfortunately, the limitations in current experimental approaches do not allow to answer quickly enough this need of validation.
In that context, we decided to develop a bioinformatics environment to optimize the construction and understanding of RNA architectures.
The first part of the present thesis is focused on the description of our infrastructure named **P.A.R.A.DIS.E** (***P**latform to **A**nalyze **R**NA **A**nnotations over a **Dis**tributed **E**nvironment*). Its development can be divided in three main axes :
– an RNA annotations (secondary and tertiary structures, sequences alignments, ...) engine
– a communication layer allowing to contact the algorithms that produce these annotations
– a graphical layer allowing to visualize and manipulate these annotations using adapted representations.
The second part is more specifically focused on the graphical tool **Assemble**, that allows to build a tertiary structure model for an RNA molecule with a known secondary structure. **Assemble** proposes a great number of automated tools speeding up the model construction by allowing to generate a first draft of the 3D model, to apply conformations derived from known structural motifs and to refine the model's coordinates according to structural constraints. The user also has the possibility to display an electronic density map to guide him during the modelling.
The third and last part is dedicated to the validation of our bioinformatics tools and describes in details their use to answer RNA-related biological questions.

Mots-Clés : ARN, développement logiciel, modélisation moléculaire, structure tridimensionnelle, analyse structurale

Remerciements

Cette thèse a été réalisée dans l'UPR 9002 «**A**rchitecture et **R**éactivité de l'AR**N**» du CNRS, dirigée par le Professeur Eric WESTHOF à l'Institut de Biologie Moléculaire et Cellulaire. Ce travail a été financé à l'aide d'un contract du ***H**uman **F**rontier **S**cience **P**rogram* (50-3284).

Je tiens d'abord à remercier le Professeur Eric WESTHOF de m'avoir accueilli dans son laboratoire. Je lui suis reconnaissant de la confiance qu'il m'a témoignée, et de m'avoir fait bénéficier de ses conseils et de son expérience.

Je remercie chaleureusement le Docteur Fabrice JOSSINET, qui m'a encadré au jour le jour. Il a su me guider dans mes travaux tout en me laissant un grand degré de liberté. Nos interactions quotidiennes ont été très stimulantes et sont une source d'inspiration pour mes reflexions scientifiques.

Je remercie le Professeur Alain DENISE d'avoir accepté de faire partie du jury de ma thèse. Ses compétences reconnues en bioinformatique des ARN et en algorithmique font que son nom s'est imposé de lui-même lors de la constitution du jury.

Je remercie le Professeur Jean-Marie WURTZ d'avoir accepté de juger mon travail de thèse. Ses travaux dans le domaine de la bioinformatique font qu'il était sans nul doute la personne la plus indiquée pour remplir la charge de rapporteur interne de cette thèse.

Je remercie le Docteur Marc DELARUE, dont les recherches dans les domaines de la cristallographie et de la modélisation moléculaire sont remarquables, d'avoir accepté de consacrer une partie de son temps à l'évaluation de cette thèse.

Je remercie l'ensemble des membres de l'équipe *Bioinformatique, modélisation et simulation d'acides nucléiques* pour les nombreux conseils et discussions.

Je remercie l'ensemble des membres de l'UPR pour leur interventions, questions et conseils lors de mes présentations orales et pour m'avoir permis de travailler dans une

ambiance agréable.

Je remercie ma famille pour m'avoir soutenu et guidé dans mes choix, me permettant ainsi d'approcher mon rêve d'enfant de devenir un jour un chercheur.

Je remercie tous mes amis Degausseurs pour leur soutient, leur bonne humeur et les conseils qu'ils m'ont prodigués durant ma thèse vis à vis de la biologie, la chimie, la programmation Java, openGL, le réseau (même si le réseau c'est mal), linux et l'informatique en général. Merci Dr Kynes, Dr Emy, Dr Em, Dr Jabba, Duncan, Tite Nélène, Dr Aktarus, Caro, Dr Gromito, Fouine (, La), Danes, AE, Dr Onclebens, Joe, Dr Dude, Iomm, Dr Poulet, Poulette, Gounok, Nashera, Benool, M27 & Yoboka.

Je remercie toutes les personnes qui m'ont apporté du soutient.

Я хочу поблагодарить мою жену, Машу, за поддержку и за одобрительные слова в течении моей докторской диссертации.
Благодарю её конечно же за то, что она согласилась провести всю свою жизнь вместе со мной. Merci Maria !
Я благодарю семью Прокудиных за тёплый приём в семью и за доверие их дочьки. Передаю привет кенгуру.

Je dédie ce travail à la mémoire de notre stagiaire Yannick Krause et de notre collaborateur québécois Martin Larose, qui nous ont quitté bien trop jeunes, durant ma thèse, et dont la disparition m'a profondément touchée.

A ma famille

Table des matières

Table des figures

Liste des tableaux

Première partie

Introduction

Chapitre 1

Introduction Générale

L'ARN (Acide RiboNucléique) a longtemps été considéré comme un simple vecteur de l'information génétique, permettant une transition entre l'ADN et les protéines par le biais des ARN messagers, ARN de transfert et ARN ribosomiques. Toutefois, l'exploitation des données produites par le décryptage des génomes a permis de mettre davantage en évidence la complexité des systèmes biologiques. On sait aujourd'hui que la complexité et le degré d'évolution d'un organisme sont directement liés à l'existence de centaines de milliers d'ARN non-codants de tailles variables [Taft07], et qui sont des éléments clés de machineries moléculaires réalisant d'importantes fonctions au sein de la cellule. De nombreuses études montrent que l'ARN participe à un nombre toujours croissant de mécanismes régulant les grandes fonctions biologiques et, par conséquent, de mécanismes les perturbant et impliqués dans les phénomènes de cancer [Zhang08, Dykxhoorn08], d'infections virales [Huang08, Kumar08] ou bactériennes [Romby06] et d'affections neurologiques [Mehler07].

Ces fonctions cellulaires importantes, ainsi que la découverte continuelle de nouveaux transcrits d'ARN de fonctions inconnues font que l'ARN connaît aujourd'hui un intérêt grandissant. Et de ce fait, on assiste depuis quelques années au développement de nombreux outils permettant l'identification de nouveaux ARN non-codants au sein des

génomes séquencés.

Comme pour toute molécule, c'est la structure d'un ARN qui va déterminer sa fonction au sein de la cellule, il est donc primordial de bien comprendre les règles qui régissent les architectures des ARN et leur à travers la phylogénie.

Pour comprendre ces règles, il est nécessaire d'avoir à disposition un nombre important de structures d'ARN. Malheureusement, leur production à haute résolution par cristallographie de rayon X ou par résonance magnétique nucléaire est relativement lente comparée au séquençage. Il en résulte un écart important entre le nombre de molécules dont on connaît la séquence et le nombre de molécules pour lesquelles la structure tridimensionnelle est connue.

Afin, de réduire cet écart, plusieurs approches aussi bien biophysiques, biochimiques que bioinformatiques ont été mise en oeuvre pour obtenir ou prédire des structures d'ARN. Avant de voir ses approches, nous allons procéder à un rappel sur l'ARN et ses propriétés.

Chapitre 2

L'ARN

L'ARN (**A**cide **R**ibo**N**ucléique) est un biopolymère formé d'une chaîne de nucléotides. L'ARN est une molécule proche de l'ADN, mais en diffère notamment par le fait que le sucre de l'ADN est un désoxyribose (un ribose auquel il manque un l'oxygène en position 2') et que l'ARN substitue généralement la base Thymine de l'ADN par l'Uracile. L'ARN est transcrit à partir de l'ADN par des enzymes appelées ARN-polymérases et est généralement traité ensuite par d'autres enzymes. Certains ARN, les ARN messager (ou ARNm) sont ensuite traduits en protéines par le ribosome (qui est lui-même en partie constitué d'ARN).

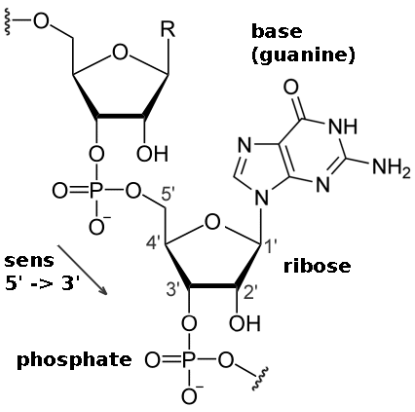


FIG. 2.1 – La structure de l'ARN
Deux nucléotides d'une chaîne d'ARN. Les atomes de Carbone du ribose sont numérotés

2.1 Les généralités structurales

Chaque nucléotide d'ARN contient un sucre ribose (**Fig. 2.2**). On numérote les atomes de carbone de ce sucre de 1' à 5'. Une base azotée (en général Adénine, Cytosine, Guanine ou Uracile : **Fig. 2.8**) est attachée en position 1'. Entre la position 3' d'un ribose et la position 5' du suivant, on trouve un groupement phosphate. La charge négative de ces phosphates fait de l'ARN une molécule chargée (polyanion).

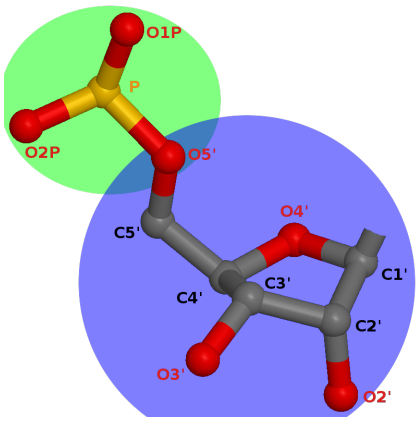


FIG. 2.2 – Le squelette Ribose-Phosphate

En bleu : le ribose, en vert le phosphate (qui sera complété par l'atome O3' du nucléotide précédent)

Une particularité structurale importante distinguant l'ARN de l'ADN est la présence d'un groupement hydroxyle en position 2' du ribose. Ceci conduit le ribose à préférer une conformation en C3'-endo alors que le désoxyribose adopte une conformation en C2'-endo. L'ARN prendra, en conséquence, une conformation de forme A plutôt que de forme B qui est plus communément observée pour l'ADN. Il en résulte que les hélices d'ARN possède un grand sillon étroit et profond, et un petit sillon large et peu profond.

Bien que l'ARN soit transcrit uniquement à l'aide de quatre bases (Adénine, Cytosine, Guanine et Uracile) il existe environ une centaine de nucléosides modifiés, dont notamment la thymidine ou la pseudouridine et l'inosine qui jouent un rôle important dans les ARN de transfert.

Le squelette sucre-phosphate

Bien que les paires de bases et les empilements soient les premiers déterminants de la structure 3D des ARN, certaines conformations du squelette sucre-phosphate sont cruciales pour l'activité catalytique des ARN, leur fixation avec des drogues ou des aptamères. Ce squelette est articulé par de nombreux angles de torsion. On trouve, le long du squelette, entre deux nucléotides successifs, sept angles de torsion (**Fig 2.4**). En raison de la variabilité des angles de torsion qui le composent, il est plus difficile d'obtenir, par des techniques de cristallographie par rayon X et de résonance magnétique nucléaire, une description au niveau atomique pour le squelette que pour les bases. De nombreuses études portent sur les conformations adoptées par le squelette sucre-phosphate de l'ARN. [Hershkovitz03, Murray03, Schneider04, Murray05, Hershkovitz06, Richardson08]

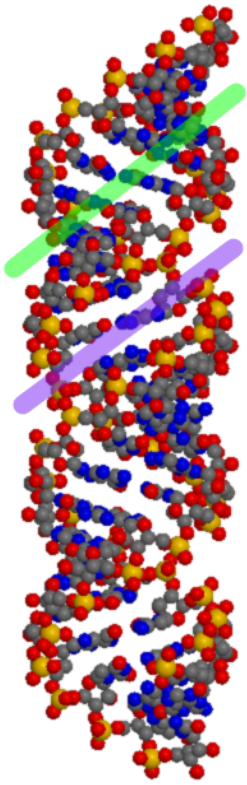
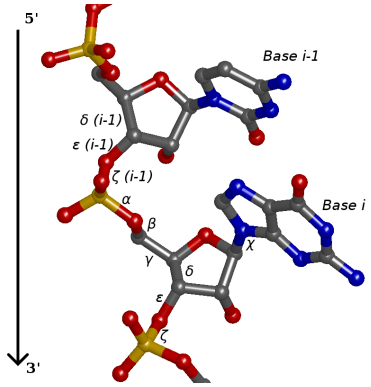


FIG. 2.3 – L'hélice d'ARN
En vert et violet, les petit et grand sillons

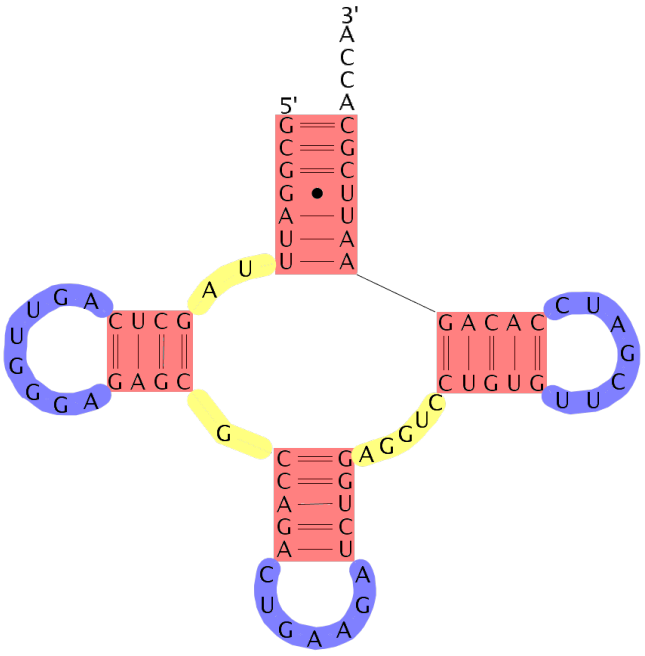


Les différents angles de torsion du squellette (α, β, γ, δ, ϵ et ζ) et l'angle χ entre le sucre et la base

FIG. 2.4 – Les angles de torsion entre deux nucléotides

2.2 La structure secondaire de l'ARN

Comme c'est le cas pour l'ADN, des liaisons hydrogène peuvent s'établir au sein du polymère d'ARN et former des paires de bases. Ainsi une guanine peut s'apparier avec une cytosine au moyen de trois liaisons hydrogène, et une adénine peut s'apparier avec une uracile grâce à deux liaisons hydrogène. Ces deux paires de bases sont dites canoniques, mais il existe d'autres types de paires, dont notamment la paire Wobble formée par une guanine et une uracile. Le réseau de ces paires de bases constitue, à plus grande échelle, des domaines structuraux tels que les hélices, les boucles en épingle à cheveux, les bulles et boucles internes. Lorsque l'on regroupe ses domaines, on obtient la structure secondaire d'un ARN. C'est cette structure secondaire qui sert de squelette au repliement tridimensionnel de l'ARN.



Cette figure représente la structure secondaire de l'ARNt^{PHE} de S.cerevisiae. En rouge, bleu et jaune on peut voir les hélices, les boucles terminales et les boucle internes. Les traits simples et doubles représentent des paires de bases canoniques, le cercle une paire wobble.

FIG. 2.5 – La structure secondaire d'un ARN

Les pseudonœuds

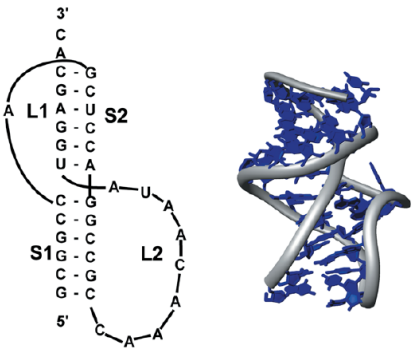


FIG. 2.6 – Un pseudonœud

Le pseudonœud du rétrovirus I SRV-I issu de la structure 1E95.(source [Staple05])

Les pseudonœuds sont des éléments structuraux d'ARN observés pour la première fois dans le *turnip yellow mosaic virus* [Rietveld82]. Ils ont une apparence de nœud dans une structure tridimensionnelle, mais n'en sont pas d'un point de vue topologique. Les pseudonœuds possèdent plusieurs types de repliements et de topologies. Dans le repliement de type H, les bases d'une boucle en épingle à cheveux forment des appariements avec des bases hors de la tige, causant la formation d'une seconde tige-boucle et d'un pseudonœud. Ces deux hélices peuvent alors s'empiler l'une au dessus de l'autre pour former une hélice quasi-ininterompue, avec un brin continu et l'autre discontinu. Certains pseudonœuds sont connus pour avoir une activité catalytique. Les introns de groupe I sont des ribozymes capables de procéder à leur propre épissage, sans recourt au complexe ribonucléoprotéique appelé spliceosome qui est en général chargé de cette tâche. L'action catalytique de ses introns est réalisée par un pseudonœud englobant les trois hélices qui le forment. [Adams04, Westhof02]. On trouve également des pseudonœuds dans la télomérase [Ulyanov07, Shefer07] et dans les domaines catalytiques de structures permettant l'hydrolyse ou la transesterification [Doudna02]. Certains ARN messagers présentent également une structure en pseudonœud en aval d'une *séquence glissante*. Ceci va ralentir le ribosome lors de la traduction et pourra potentiellement provoquer un décalage de cadre de lecture, conduisant à la synthèse d'une protéine alternative. Cette «erreur» de lecture ne se réalisant qu'en faible proportion, un même ARN messager pourra être traduit en deux protéines. [Namy06, Hansen07].

[Staple05] présente un rappel des différents types de pseudonœuds et leurs occurrences dans des structures connues.

2.3 Les alignements structuraux

Comme la structure tridimensionnelle d'un ARN dicte sa fonction, elle est grandement conservée au cours de l'évolution. La structure secondaire constituant l'échafaudage de cette structure tertiaire, il est logique qu'elle soit, à un certain degré, également conservée. On a observé au travers des données génomiques, que la séquence primaire n'est, quant à elle pas autant conservée. La structure secondaire est composée d'hélices, dont les éléments unitaires sont les paires de bases canoniques. Une paire de bases canonique peut se substituer à une autre (par mutation des deux bases qui la composent) au sein d'une hélice sans en perturber la géométrie.
Repérer ces covariations au sein de séquences d'ARN réalisant des fonctions analogues et appartenant à des organismes différents permet de mettre en lumière les conservations et évolutions structurales de cette famille de molécules. Cela peut se faire par l'alignement des séquences entre elles. Comme on considère l'information de structure en plus des séquences brutes, il s'agit d'un alignement structural.

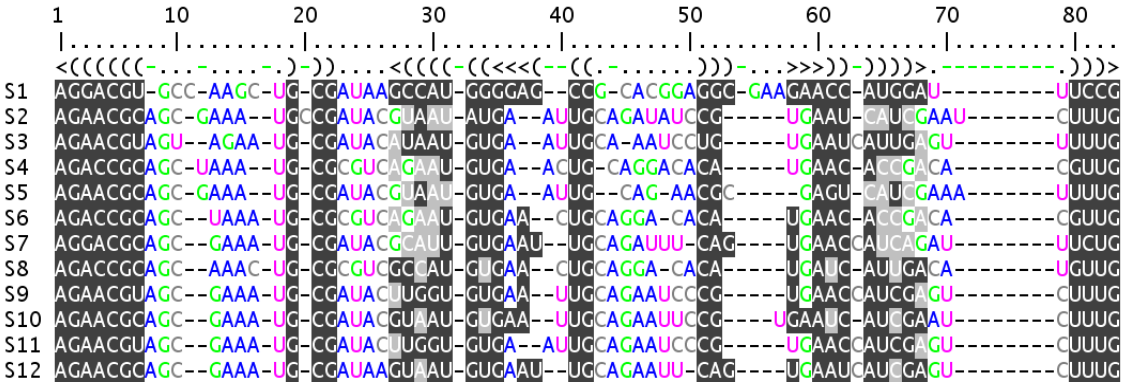


FIG. 2.7 – Un alignement structural

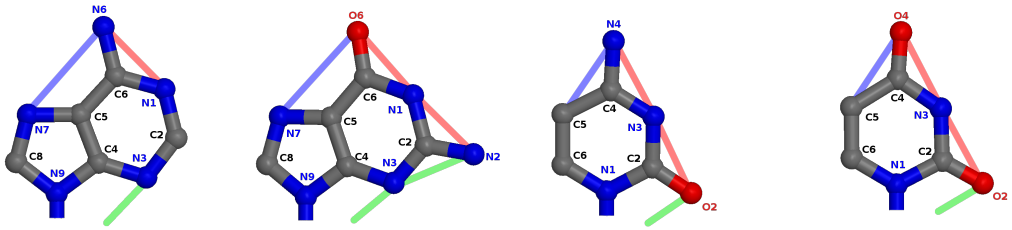
Pour une famille d'ARN donnée, chaque ligne contient la séquence d'un orthologue. Les nucléotides alignés verticalement sont considérés comme occupant des positions structuralement équivalentes

Toutefois, pour comprendre pleinement la structure tertiaire d'un ARN, on ne peut se limiter à la prise en compte des seules interactions canoniques. En effet, il existe un nombre important d'interactions possibles entre les quatre bases azotées de l'ARN [Hoogsteen63, Saenger84] et c'est le réseau de ces interactions qui va véritablement dicter les particularités intrinsèques de chaque structure d'ARN. L'étude de ces interactions à conduit à la mise en place d'une classification.

2.4 La classification Leontis-Westhof

Chaque base possède trois faces à l'aide desquelles elle peut former des interactions au moyen de liaisons hydrogènes avec une autre base. Ces faces sont :

- la face Watson-Crick (par laquelle se forme entre autre les paires de bases canoniques et wobbles)
- la face Hoogsteen
- la face Sugar Edge

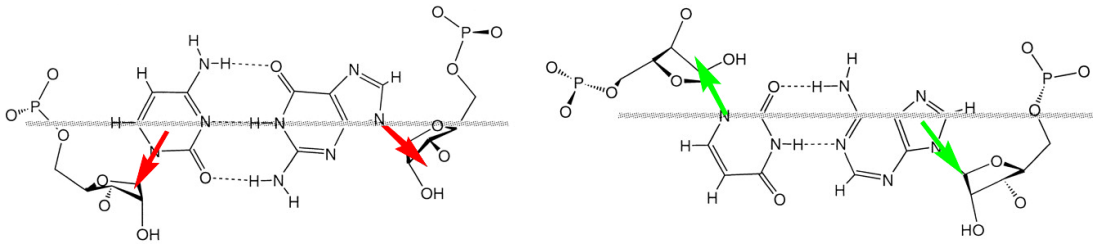


(a) Adénine (b) Guanine (c) Cytosine (d) Uracile

FIG. 2.8 – Les bases de l'ARN

A gauche : les bases puriques, à droite : les bases pyrimidiques. Les lignes rouges, bleues et vertes représentent les faces Watson-Crick, Hoogsteen et Sugar-Edge. La face Sugar-Edge implique également le groupement hydroxyle O2' du ribose

Deux faces données peuvent former ensemble des interactions différentes en fonction de leur orientation relative. Comme il existe trois faces, il existe 6 combinaisons entre ces faces pouvant chacune adopter l'orientation cis ou trans. On obtient donc 12 types d'interactions possibles entre les bases de l'ARN (voir tableau **2.1**).



(a) Orientation cis

(b) Orientation trans

FIG. 2.9 – L'orientation de l'interaction par rapport à la liaison glycosidique

Lorsque les liaisons glycosidiques des deux nucléotides formant l'interaction sont du même coté des liaisons hydrogènes, l'orientation de cette interaction est cis, dans le cas contraire, elle est trans

Chacun de ces types d'interactions va localement modifier l'orientation relative entre les deux brins d'ARN porteurs des bases. La classification des paires de bases à également conduit à la mise en place d'une nomenclature relative à ces interactions : la nomenclature *Leontis-Westhof* [Leontis01]. Dans cette nomenclature, chaque face est représentée par un symbole géométrique : un cercle pour la face Watson-Crick, un carré pour la face Hoogsteen et un triangle pour la face Sugar Edge. L'orientation de l'interaction est, quant à elle, représentée par un remplissage plein pour l'orientation cis et creux pour l'orientation trans.

N°	Orientation de la liaison glycosidique	Faces impliquées	Orientation locale des brins	Symbole
1	cis	Watson-Crick/Watson-Crick	Antiparallèle	
2	trans	Watson-Crick/Watson-Crick	Parallèle	
3	cis	Watson-Crick/Hoogsteen	Parallèle	
4	trans	Watson-Crick/Hoogsteen	Antiparallèle	
5	cis	Watson-Crick/Sugar Edge	Antiparallèle	
6	trans	Watson-Crick/Sugar Edge	Parallèle	
7	cis	Hoogsteen/Hoogsteen	Antiparallèle	
8	trans	Hoogsteen/Hoogsteen	Parallèle	
9	cis	Hoogsteen/Sugar Edge	Parallèle	
10	trans	Hoogsteen/Sugar Edge	Antiparallèle	
11	cis	Sugar Edge/Sugar Edge	Antiparallèle	
12	trans	Sugar Edge/Sugar Edge	Parallèle	

TAB. 2.1 – Les 12 familles d'interactions et leurs symboles

La classification *Leontis-Westhof* et sa nomenclature associée permettent donc d'annoter facilement les structures tertiaires d'ARN et de mettre en valeur les différents réseaux d'interactions.

2.5 L'isostérie entre paires de bases

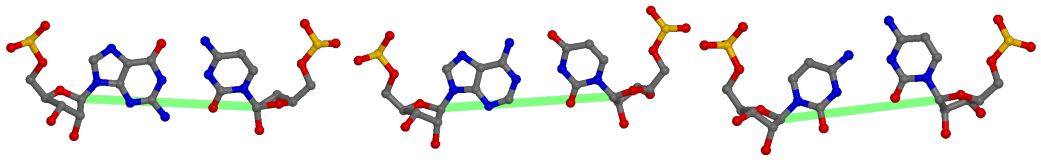
La structure tridimensionnelle des molécules homologues d'ARN change beaucoup plus lentement que la séquence de ces molécules au cours de l'évolution. Par définition, des molécules homologues partagent une origine biologique et une fonction commune. Des mutations aléatoires, à des points structurellement importants des molécules d'ARN, sont accommodées par la sélection naturelle lorsqu'elle perturbent peu la structure locale ou lorsqu'elles sont compensées par des mutations en d'autres points de la séquence. L'observation de ces covariations sur des interactions de type cis Watson-Crick/Watson-Crick, a permis de prédire avec succès la conservation de doubles hélices dans des molécules d'ARN homologues. C'est l'isostérie entre les paires de bases canonique A-U, C=G, G=C et U-A qui est à la source de cette propriété. En effet, les distances C1'-C1' dans chacune de ces paires sont identiques.

Deux paires de bases a-b et c-d sont dites ***isostériques*** [Leontis02b] si :

- les faces impliquées sont les même entre a et c et entre b et d
- l'orientation de l'interaction par rapport à la liaison glycosidique est la même
- lorsqu'on substitue a par c et b par d, on ne perturbe pas la géométrie locale (la distance entre les atomes C1' des deux partenaires reste fixe)

Il est à noter que si la distance C1'-C1' est égale (ou quasi égale) il n'en est pas de même pour le volume global occupé par les bases.

Les paires de bases appartenant à une même famille géométrique (par exemple cis Watson-Crick/Watson-Crick) montrent des orientations relatives de leurs liaisons glycosidiques très similaires, ce qui implique la conservation locale de l'orientation des brins. Toutefois, toutes les paires de bases d'une famille géométrique ne sont pas isostériques les unes par rapport aux autres, car la distance entre les C1' peut varier d'une paire à l'autre (voir la figure **2.10**).



(a) Une paire G-C (10.3 Å) (b) Une paire A-U (10.3Å) (c) Une paire C-C (8.5Å)

FIG. 2.10 – Un exemple de paires isostériques et non-isostériques

Trois paires de bases cis Watson-Crick/Watson-Crick et leurs distances C1'-C1' associées. (a) et (b) sont isostériques, mais ne sont pas isostériques à (c)

Les matrices d'isostérie

Cette distance a donc été utilisée comme discriminant pour diviser les familles géométriques en sous-familles isostériques. Ces sous familles permettent d'identifier

des paires de bases pouvant se substituer à d'autres par covariation tout en préservant la structure tridimensionnelle. Cette information est capitale pour la modélisation moléculaire, la prédiction de motifs structuraux et de structure tertiaire, et pour l'obtention de meilleurs alignement structuraux. Le partitionnement d'une famille géométrique en sous familles isostériques peut être représenté par une matrice d'isostérie (voir tableau **2.2**). Au sein d'une telle matrice, toutes les paires de bases portant le même label sont isostériques. Les paires n'ayant pas de label, n'ont jamais été observées au sein de structures cristallographiques.

cis	**A**	**C**	**G**	**U**
A	I_4	I_2	I_3	I_1
C	I_2	I_6	I_1	I_5
G	I_3	I_1		I_2
U	I_1	I_5	I_2	I_6

TAB. 2.2 – La matrice d'isostérie de la famille cis Watson-Crick/Watson-Crick

Cette famille géométrique se partitionne en 6 sous familles d'isostérie. Notons que la famille I_1 correspond aux appariements canoniques formant les doubles hélices d'ARN. La paire G-G n'a, à ce jour, jamais été observée en structure cristallographique.

2.6 Les motifs structuraux d'ARN

Les structures tridimensionnelles d'ARN présentent des réseaux d'interactions complexes qui, pour une grande partie, reposent sur des paires de bases non canoniques et forment des motifs structuraux [Westhof04, Krasilnikov04, Leontis03]. Ces motifs, observés de façon récurrente, notamment dans les ARNr [Wimberly00, Ban00], indépendamment de la fonction et de l'organisme de l'ARN observé, favorisent les interactions ARN-ARN à longue distance et créent des sites de fixation pour des protéines et de petits ligands. La conservation de tels motifs structuraux est bien plus grande que celle de leurs structures secondaires sous-jacentes, car plusieurs structures secondaires peuvent former des repliements tridimensionnels similaires.

On peut définir un motif par une liste ordonnée de paires de bases non canoniques

pouvant être interrompues par des bases insérées ou en bulle. Ces motifs sont souvent inclus dans des régions hélicales, formant ainsi des boucles internes ou des épingles à cheveux.

2.6.1 Un exemple de motif structural : le Kink-Turn

Les motifs Kink-Turn [Klein01] sont des boucles internes récurrentes, qui produisent une courbure importante dans les hélices d'ARN (**Fig. 2.12**).

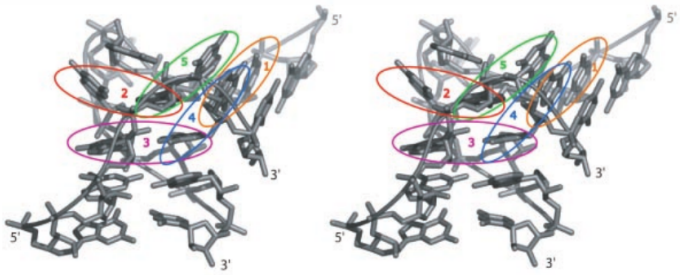


FIG. 2.12 – Le Kink-Turn en 3D

Une vue stéréographique du motif Kink-Turn tel qu'il est observé dans la structure cristallographique du ribosome de H.marismortui

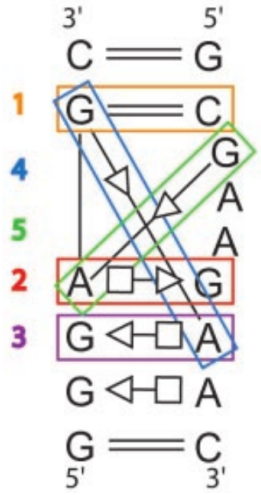
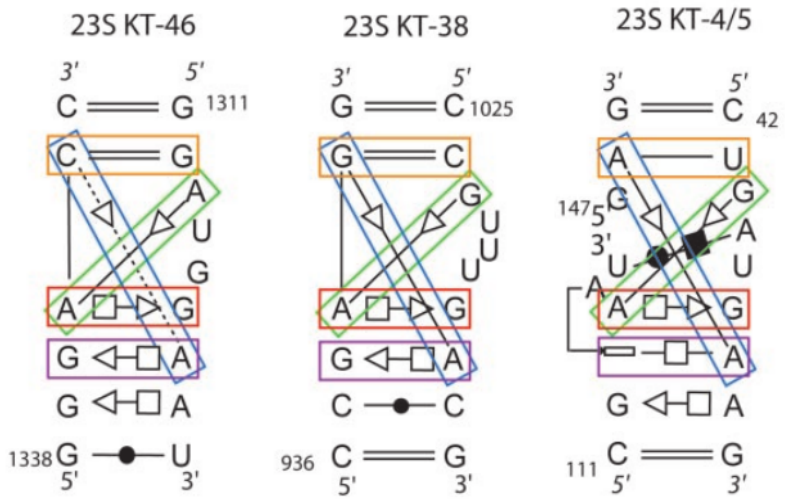


FIG. 2.11 – Le Motif Kink-Turn

Les cinq paires de bases formant le motif sont colorées et numérotées

Cette courbure rapproche les petits sillons des deux hélices qui encadrent le motif. De nombreuses études théoriques et expérimentales ont permis de mettre en valeur la flexibilité des Kink-Turn, ainsi que leur besoin en magnésium et protéines [Goody04, Matsumura03, Cojocaru05, Rázga04]. Le motif du Kink-Turn, caractérisé par cinq paires de bases, peut être représenté grâce à la nomenclature *Leontis-Westhof* (**voir figure 2.11**). Des Kink-Turn ont pu être observés dans des structures aussi variées que la snRNP U4 [Vidovic00], la pseudouridine synthétase à boîte H/ACA [Rozhdestvensky03], les méthylases à boîte

C/D [Kuhn02] ou l'élément autorégulateur de l'ARNm L1 [Nevskaya05]. Plusieurs combinaisons d'interactions pouvant adopter le même repliement tridimensionnel, on a observé dans les structures cristallographiques de nombreuses variantes du Kink-Turn qui correspondent aux schémas de la figure **2.11**. Ces variantes ne sont pas exhaustives et plusieurs versions sont présentées dans [Lescoute05]. On peut également noter l'existence d'un motif Kink-Turn inversé [Strobel04].



(a) Isostérie (b) Insertion de base (c) Restructuration

FIG. 2.13 – Des versions alternatives du Kink-Turn

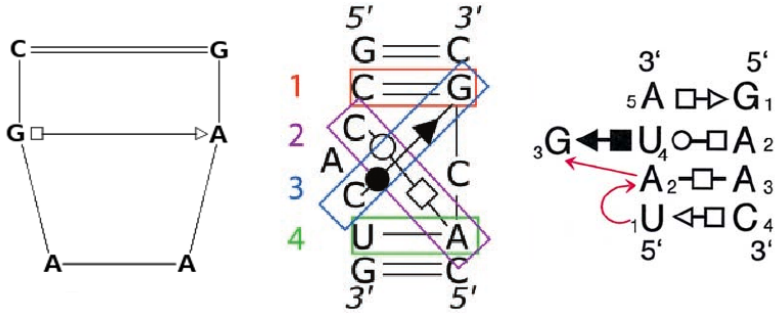
*Ces trois versions du Kink-Turn différent de la version présentée en figure **2.11** : (a) Les paires 1, 4 et 5 sont remplacées par des paires isostériques. (b) Une base est insérée dans la bulle entre les paires 5 et 2. (c) Une restructuration importante des paires de bases permet de former le même motif en 3D*

2.6.2 D'autres motifs

Hormis le Kink-Turn il existe de nombreux motifs structuraux d'ARN dont voici quelques exemples :

- **la boucle GNRA** : dans certaines molécules les boucles avec une séquence GNRA (ou R est une purine et N un nucléotide quelconque) représentent plus d'un tiers des boucle à quatre nucléotides. Ce motif structural, relativement stable énergétiquement, est connu pour interagir avec le petit sillon d'une hélice distante [Jaeger94, Pley94, Costa97, Costa95]. Le motif GNRA intervient également dans des interactions boucle-boucle [Lehnert96] [Costa97] [Costa00]. On observe notamment ce motif dans les introns de groupe I [Lehnert96] et le ribozyme à tête de marteau [Jucker95]
- **la C-loop** : ce motif est une boucle interne asymétrique qui augmente la torsion hélicale d'une hélice d'ARN, permettant à sa boucle terminale d'établir des interactions tertiaires [Lescoute05]. On retrouve ce motif, entre autres, dans les ARN 23S [Ban00] et 16S [Clemons01, Wimberly00] et l'ARNm de la thréonine synthétase [Torres-Larios02]
- **Sarcin-Ricin** : ce motif est un empilement de paires de bases non canoniques servant de site pour des interactions entre l'ARN et des protéines ou d'autres ARN [Leontis02a]. Ce motif a notamment été observé dans plusieurs structures cristallographiques des petites et grandes sous unités du ribosome [Ban00, Wimberly00, Schluenzen00, Harms01]

La représentation par schéma utilisant la nomenclature *Leontis-Westhof*, associée aux matrices d'isostérie des différentes paires de bases, permet de mettre en valeur les covariations menant à la conservation ou à l'évolution des motifs structuraux d'ARN à travers la phylogénie. De plus, l'étude et la mise en valeur de motifs structuraux d'ARN joue un rôle important dans le processus d'affinement des alignements structuraux et de prédiction de structures tertiaires.



(a) GNRA (b) C-loop (c) Sarcin-Ricin

FIG. 2.14 – Quelques motifs structuraux

Des études approfondies des motifs structuraux d'ARN sont présentées dans [Lescoute05, Lescoute06a, Wexler07]

Chapitre 3

L'analyse bioinformatique des architectures ARN

L'analyse structurale des ARN non codants peut s'effectuer à plusieurs niveaux et en utilisant des approches très variées. De nombreuses techniques permettant d'inférer de l'information structurale concernant les ARN sont notamment issues de la biochimie (sondage chimique ou enzymatique, empreinte chimique ou enzymatique, mutagénèse dirigée ou aléatoire, ...) et de la biophysique (cristallographie par diffraction de rayons X, résonance magnétique nucléaire, cryo-microscopie électronique, ...) [Felden07]. Dans le cadre de cette thèse, nous ne détaillerons pas ces techniques. L'alternative à ces approches que nous avons considérée est l'approche bioinformatique (ou *in silico*).

 La **bioinformatique** est un champ de recherche multidisciplinaire regroupant, entre autres, l'informatique et les mathématiques appliquées dans le but de répondre à des problèmes scientifiques posés par la biologie.

La bioinformatique est très largement utilisée pour des problématiques d'analyse de séquences, d'annotation de génomes, d'analyse de l'expression génétique et protéique,

de la génomique comparative... Dans le contexte de cette thèse, nous nous intéressons surtout à son utilisation dans le cadre de l'étude des ARN non codants et notamment dans leur étude structurale.

Une molécule d'ARN est entre autres caractérisée par :

- sa séquence primaire
- sa structure secondaire
- sa structure tertiaire
- sa famille fonctionnelle
- les motifs structuraux qui la composent
- ses partenaires dans la cellule

Il existe donc de nombreux sous domaines de la bioinformatique des ARN, dont le but est de faire le lien entre ces différentes informations (qui ne sont malheureusement pas toujours toutes connues pour chaque ARN) afin de mieux comprendre les particularités structurales des ARN ainsi que leurs évolutions. Ses approches, résumées dans le schéma de la figure **3.1**, seront détaillées et illustrées par un certain nombre de techniques et d'outils les exploitant.

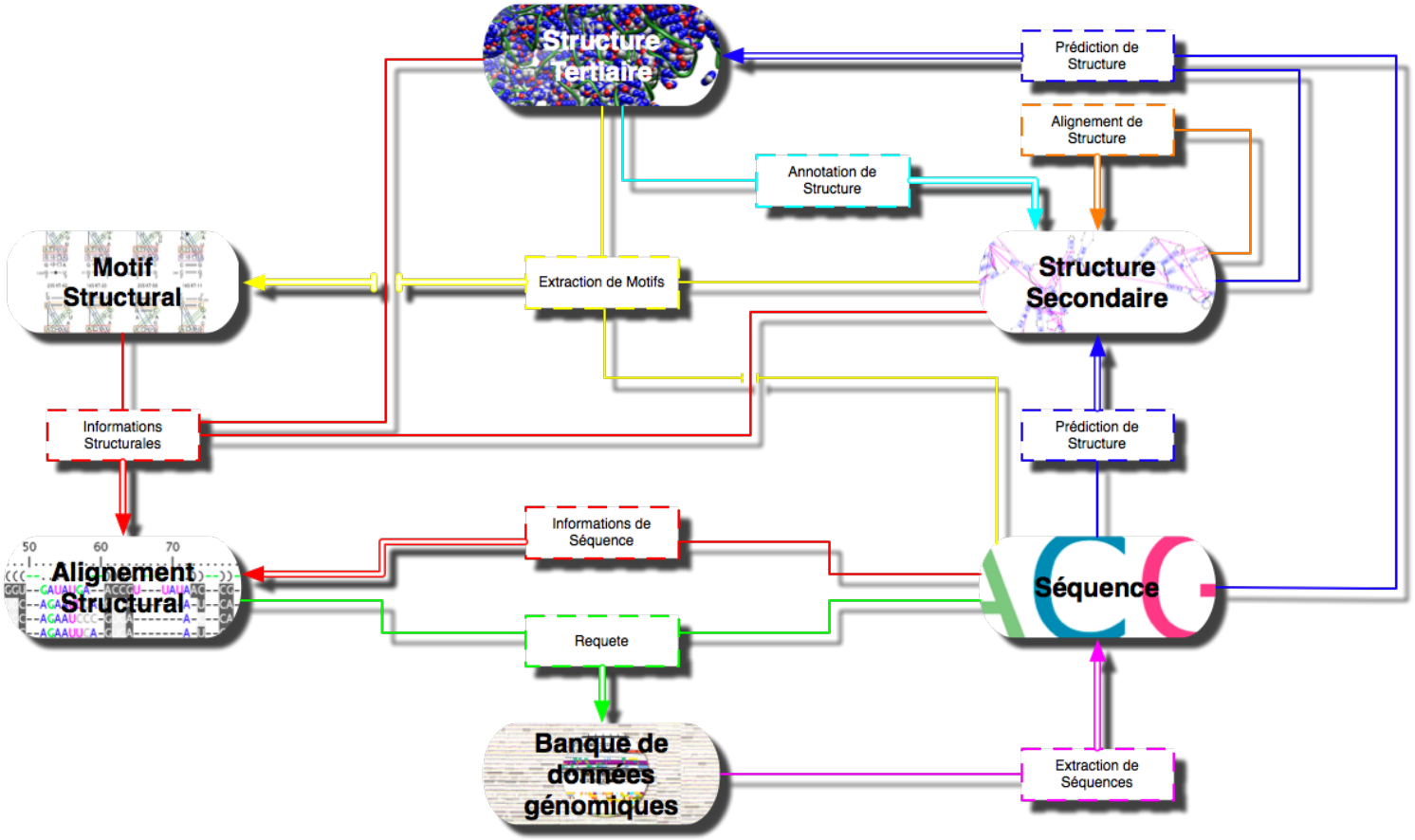


FIG. 3.1 – Les études structurales *in silico* de l'ARN

Algorithmes & Heuristiques

Un **algorithme** est une séquence d'instructions, définie dans le but de produire un résultat à partir d'un état initial. Dans le cadre des problèmes d'optimisation (ex : «rechercher la structure secondaire de plus petite énergie libre»), l'algorithme à pour but de trouver la meilleure solution.

Une **heuristique** est une méthode de résolution de problèmes d'optimisation qui, par le biais de simplifications du problème, permet de trouver rapidement une solution sous-optimale mais satisfaisante du problème.

Quelques mots sur la complexité des algorithmes

On note par $O(x)$ la complexité d'un algorithme en fonction de la quantité n de données à traiter. Il s'agît du nombre d'opérations élémentaires à exécuter pour mener l'algorithme à terme. Dans les programmes décrits dans ce chapitre, n représente le nombre total de nucléotides à traiter. Le tableau **3.1** montre le temps d'exécution d'un programme en fonction de sa complexité et de la quantité de données sur un ordinateur nécessitant une micro seconde par opération élémentaire.

n	$O(n)$	$O(n^2)$	$O(n^3)$	$O(n^6)$	$O(2^n)$
1	1 μs	1 μs	1 μs	1 μs	2 μs
10	10 μs	100 μs	1 ms	1 s	1 ms
100	100 μs	10 ms	1 s	12 jours	10^{16} ans
1000	1 ms	1 s	16 mn	32000 ans	10^{287} ans
10000	10 ms	100 s	12 jours	10^{10} ans	10^{3000} ans

TAB. 3.1 – Approximation des temps d'exécution des algorithmes

Avec la séquence d'une molécule d'ARN, il peut être intéressant de connaître la structure tridimensionnelle (ou à défaut la structure secondaire) qui lui est associée. La bioinformatique propose plusieurs approches pour répondre à cette question, lorsque les approches de biologie expérimentale s'avèrent trop coûteuses.

3.1 Les alignements de molécules d'ARN

L'alignement de séquences est une méthode visant à arranger les nucléotides de manière à mettre en valeur des régions présentant des similarités ou dissemblances qui sont le fruit de l'évolution. Ainsi, si deux séquences d'un alignement ont un ancêtre commun, les zones dissemblables sont le reflet de mutations, d'insertions ou de délétions. Lorsqu'un jeu de séquences est correctement aligné, les éléments alignés verticalement (nucléotides ou séries de nucléotides) représentent des régions homologues entre les différentes séquences. L'approche *a priori* la plus appropriée pour aligner deux séquences A et B, est d'utiliser la programmation dynamique afin de trouver le nombre minimal d'éditions (mutations, insertions ou délétions) pour transformer A en B. Toutefois, comme nous allons le voir, cette approche ne reflète par forcement le processus naturel d'évolution entre les séquences, et notamment l'évolution des séquences d'ARN.

Programmation dynamique

La programmation dynamique est une méthode de résolution de problèmes d'optimisation qui consiste à déduire la solution optimale d'un problème à partir de solutions optimales de sous problèmes le composant. Cet approche consiste donc à :

1. décomposer le problème en sous problèmes plus petits
2. trouver la solution optimale de chaque sous problème (en utilisant également cette méthode de programmation dynamique)
3. utiliser ces solutions optimales de sous problèmes pour établir la solution optimale du problème original.

3.1.1 Les alignements de séquences

`CLUSTAL W` [Thompson94] est encore aujourd'hui l'un des outils les plus utilisés pour les alignements de séquences de protéines ou d'acides nucléiques. L'alignement se déroule en trois étapes : d'abord toutes les séquences sont alignées deux à deux et classées par similarité. Ensuite, un arbre-guide est construit à partir de la matrice de distance obtenue à la première étape. Enfin les séquences sont réalignées progressivement, dans un ordre déterminé par l'arbre.

`Opal` [Wheeler07] aligne également l'ensemble des séquences grâce à un alignement progressif de sous-alignements.

[Colbourn07] aligne les séquences trois par trois et intègre ces résultats dans un alignement multiple.

`Align-m` [Van Walle04] est adapté à l'alignement de séquences qui sont très dissemblables, grâce à l'usage de différentes matrice de scores.

`MSA-GA` [Gondro07] utilise un algorithme génétique et fait évoluer une population d'alignements candidats pour obtenir l'alignement avec le meilleur score en un temps d'exécution rapide. La qualité de chaque alignement est déterminée uniquement à partir des informations de séquence.

`MUMmerGPU` [Schatz07] permet d'aligner parallèlement chacune des séquences en entrée contre une séquence de référence. La parallélisation de ces alignements est rendue possible en faisant usage de l'architecture *CUDA* de nVidia, permettant de faire exécuter les calculs par des processeurs graphiques, ce qui rend le programme 3.5 fois plus rapide que son équivalent utilisant le processeur principal de l'ordinateur [Delcher02]. L'alignement de deux séquences se fait en construisant un arbre de suffixes représentant chaque séquence, et en alignant ces deux arbres.

De nombreux exemple de programmes d'alignements sont disponibles à l'adresse [alignements]. Ce type d'approches n'est basé que sur les informations de séquence et a surtout été utilisé pour aligner des génomes et des protéines. Comme nous l'avons vu, les séquences d'ARN sont bien moins conservées que les structures secondaires et

tertiaires qu'elles forment. L'intégration d'informations de structure dans un algorithme d'alignement de séquences d'ARN représente donc un atout majeur et reflète bien mieux la réalité biologique de ces molécules.

3.1.2 Les alignements structuraux

En général, l'analyse comparative d'une famille de séquences homologues permet de mettre en valeur des covariations au sein de cette famille d'ARN et de définir la position des hélices. Toutefois, déterminer le cœur conservé de structure secondaire dans un alignement de séquences est un travail délicat et itératif, généralement fait «à la main». Plusieurs outils facilitent ce processus en liant visuellement l'alignement à des contraintes structurales. Ces outils se concentrent sur le respect des paires de bases canoniques et de l'isostérie (**`Ribostral`** [Mokdad06], **`S2S`** [Jossinet05] et **`Sarse`** [Andersen07]) ou affichent une représentation de la structure secondaire/tertiaire pour une ou plusieurs séquences (**`4SALE`** [Seibel06], **`S2S`** [Jossinet05] et **`Sarse`** [Andersen07]). **`ConStruct`** [Wilm08] est un outil graphique semi automatique pour produire des alignements de séquences. Ce programme propose un premier alignement et aide l'utilisateur à le corriger grâce à des informations de thermodynamique et de covariation.
Dans [Gardner05], différents outils d'alignement structural de molécules d'ARN sont comparés.

3.2 La prédiction de structures secondaires

L'alignement structural de séquences d'ARN et la recherche d'une structure secondaire consensus pour ces séquences sont des problèmes très proches et de nombreux outils les traitent de façon simultanée.

3.2.1 Prédiction de structures & alignement de séquences

Plusieurs outils cherchent à produire un alignement structural et à prédire la structure secondaire des molécules simultanement. Les outils **SCARNA** [Tabei06], **STRAL** [Dalli06], **StructMiner** [Yang04] et **MARNA** [Siebert05] prédisent une structure secondaire pour chaque séquence, et cherchent ensuite à aligner ces structures. Malheureusement, le repliement individuel de chaque séquence, sur la base de l'énergie, ne garantit pas de produire la structure secondaire fonctionnelle (voir **3.2.3**).
D'autres outils se basent sur l'algorithme de Sankoff [Sankoff85] et propose de prédire la structure des ARN et de les aligner simultanément (**FOLDALIGN** [Havgaard05, Havgaard07], **STEMLOC** [Holmes05] ou **Dynalign** [Mathews02, Mathews05, Harmanci07]).
foldalignM [Torarinsson07] couple les matrices de probabilités d'appariement de McCaskill [McCaskill90] et celles de **FOLDALIGN** pour aligner les séquences. Ce programme est également capable de partitionner les séquences et de fournir un alignement pour chaque partition.
En raison de la complexité de ce type d'algorithme, de nombreux logiciels analogues imposent des limitations pour rendre les calculs réalisables. Il est par exemple fréquent que de tels outils se limitent à l'alignement de deux séquences.(**SCARNA**, **STEMLOC**, **FOLDALIGN** et **Dynalgin**). **MARNA** limite la somme de la longueur de l'ensemble des séquences à 10 000 nucléotides. **STRAL** tente de passer outre ce type de limitation en employant une heuristique. **MXSCARNA** [Tabei08] procède à un alignement multiple de plusieurs séquences d'ARN de la même manière que son prédécesseur (**SCARNA**) aligne deux séquences. Les séquences sont alignées deux à deux et un arbre phylogénétique est établi par la méthode UPGMA (*Unweighted Pair Group Method with Arithmetic mean*) pour fusionner les différents alignements en un unique alignement multiple. **aln3nn** [Kruspe07] se base sur des alignements trois par trois. **Murlet** [Kiryu07] utilise une variante de l'algorithme de Sankoff pour produire un alignement de séquences d'ARN. Une simplification de la fonction de coût permet d'utiliser moins de ressources

informatiques tout en restant efficace. La prédiction de la structure secondaire consensus est déléguée à un outil externe. Pour réduire le fort coût en ressources informatiques, des outils travaillent sur des abstractions de structures secondaires, telles que les arbres hiérarchiques de cycles **RSMATCH** [Liu05b].

Une autre stratégie consiste à aligner des séquences d'ARN non repliées en accord avec une molécule de référence, pour laquelle on connaît la structure secondaire décrite à l'aide d'un environnement probabiliste. Un environnement probabiliste est une boîte à outils statistique qui évalue l'association d'un label à une séquence linéaire (ces labels pouront par exemple être «gène», «intron» et «exon») [Eddy96, Eddy98, Eddy04]. Parmi ces environnements, les modèles de Markov cachés (HMM) ont été adaptés avec succès à la prédiction de gènes. Toutefois, ils ne sont pas capables de tenir compte d'interactions entre deux symboles distants, comme les paires de bases de l'ARN. En représentant les ARN sous forme d'arbres, le groupe de Sakakibara propose d'aligner cet arbre contre une séquence d'ARN à l'aide d'une extension des HMM permettant de gérer les pseudonœuds [Matsui05], ou plus récemment à l'aide de champs aléatoires conditionnels (Conditional Random Fields) [Sato05].

[Song06] utilise une extension des grammaires probabilistes non contextuelles (*SCFG : Stochastic context-free grammars*) pour aligner des séquences d'ARN et en prédire une structure secondaire consensus. Une amélioration de l'algorithme d'alignement par rapport à la précédente version [Cai03] permet de baisser sa complexité de O(n^4) à O(n^2).

MASTR [Lindgreen07] utilise une méthode de Monte Carlo pour produire un alignement initial aléatoirement. Une méthode de recuit simulé est alors utilisée pour corriger l'alignement et obtenir la structure secondaire consensus. **MASTR** peut également tenir compte de contraintes d'appariement ou de non appariement pour des nucléotides donnés.

RNA Sampler [Xu07] utilise les probabilités d'appariement de chaque base pour produire des structures secondaires pour les séquences en entrée. Des scores d'alignement entre les bases des différentes séquences sont utilisés pour aligner les

séquences en accord avec les structures. De façon itérative, cet alignement est utilisé pour corriger les structures secondaires, qui sont utilisées pour affiner l'alignement. Ceci assure une convergence vers un alignement correct et une structure secondaire consensus.

Méthode de Monte Carlo

La méthode de Monte Carlo est une heuristique qui utilise un échantillonage aléatoire répété pour produire son résultat. Elle s'applique quand le domaine de recherche de l'algorithme est trop vaste pour trouver une solution en testant toutes les possibilités dans un temps raisonnable. La méthode de Monte Carlo suit généralement les étapes suivantes :

1. Définir le domaine des solutions possibles
2. Générer un grand nombre de ces solutions potentielles et les évaluer
3. Faire la synthèse de tous ces résultats pour en déduire une solution sous-optimale du problème

3.2.2 La prédiction de structures secondaires à partir d'alignements de séquences

De nombreux algorithmes prédisent la structure secondaire consensus d'un alignement de séquences sur la base de l'information mutuelle. Toutefois des séquences trop proches n'offrent que peut d'informations de covariations, et des séquences trop différentes sont difficilement alignables, rendant l'information mutuelle difficile à dégager du bruit.

[Engelen07] propose un algorithme permettant de mesurer l'aptitude d'un jeu de séquence à être utilisé dans une approche comparative.

`RNAalifold` [Hofacker07] du **`Vienna RNA package`** prédit une structure secondaire consensus pour un alignement de séquences en minimisant l'énergie libre globale et en utilisant l'information de mutations compensatoires issue de l'alignement.

L'outil **KNetFold** [Bindewald06] repose, quant à lui, sur une méthode d'apprentissage automatique. Pour chaque pair de colonnes de l'alignement, **KNetFold** calcule un score de thermodynamique, de possibilités d'appariements et de covariations. **KNetFold** utilise la méthode d'apprentissage supervisé des k plus proches voisins pour déduire une unique structure secondaire (pouvant inclure des pseudonœuds) de la matrice de score précédemment établie.

HXMATCH [Witwer04] utilise une combinaison de scores de thermodynamique et de covariation. Ce programme produit deux structures secondaires sans pseudonœud qui, une fois superposées, permettent d'obtenir une structure secondaire avec pseudonœuds.

Lorsqu'un alignement de séquences d'ARN est disponible, **Mifold** [Freyhult05] permet d'identifier les sites de covariation en utilisant l'information mutuelle. Cette information est exploitée à l'aide de l'algorithme de Nussinov [Nussinov78] pour établir une structure secondaire sans pseudonœud. Le score des bases impliquées dans des paires est remis à zéro et une deuxième structure secondaire est calculée. De manière analogue à **HXMATCH** ces deux structures secondaires permettent de déduire une structure secondaire avec pseudonœuds.

Le programme **ILM** [Ruan04] utilise également une combinaison de scores de thermodynamique et de covariation, traitée par l'algorithme de Nussinov pour déterminer une structure secondaire avec pseudonœuds.

Lorsque l'on ne dispose pas d'un grand nombre de séquences que celles-ci sont tellement semblables qu'aucune information de covariation ne peut être tirée d'un alignement, il est nécessaire de disposer d'outils capables de prédire la structure secondaire d'une molécule d'ARN, sur la base de sa séquence uniquement.

3.2.3 La prédiction de structures secondaires à partir d'une séquence

La prédiction de la structure secondaire d'un ARN à partir de sa seule séquence est l'un des domaines de recherche les plus anciens de la bioinformatique des ARN, et c'est une problématique qui est encore loin d'être close aujourd'hui. Il existe plusieurs méthodes pour y parvenir. La plus ancienne utilise des paramètres thermodynamiques et se base sur le fait qu'à une température donnée, une séquence d'ARN va avoir tendance à se replier en une même structure, énergétiquement plus stable que les autres.

3.2.3.1 Les approches utilisant la programmation dynamique

Les programmes de prédiction de structures secondaires les plus utilisés reposent sur un algorithme de programmation dynamique. Ainsi, **`MFOLD`** [Zuker03], **`RNAFold`** [Hofacker03] et **`RNAstructure`** [Mathews99, Mathews04b] utilisent une approche déterministe qui leur garantit de trouver la structure de plus petite énergie libre, en accord avec le modèle énergétique qu'ils emploient. Cette approche leur permet également de proposer une groupe de structures énergétiquement sous-optimales. Par ailleurs, la dernière version de RNAstructure autorise l'utilisateur à fixer certaines contraintes en utilisant des données issues d'expériences de sondage chimique. La qualité de la prédiction d'une structure secondaire d'ARN par minimisation de l'énergie libre dépend directement de la qualité du modèle énergétique utilisé. Une estimation précise des paramètres de ce modèle est donc primordiale [Mathews99, Andronescu07]. La limitation de cette approche réside également dans le fait que, l'ARN étant fléxible et dynamqie, il n'adopte pas toujours le repliement dicté par la maximisation des paires Watson-Crick et de leurs empilements. Il s'agit donc de chercher, parmi les structures énergétiquement sous-optimales, celle pour laquelle le pourcentage de paires de bases correctement prédites est le plus élevé. L'une des solutions consiste à prédire la probabilité d'occurrence de chaque paire de bases [Mathews04a].

L'espace des solutions des algorithmes de programmation dynamique est relativement vaste, et il est nécessaire de le réduire pour que les temps de calcul soient raisonnables. Le programme **SFold** [Ding05, Chan05] échantillonne les structures sous-optimales en utilisant la probabilité de distribution de Boltzmann. Cette méthode permet de produire une structure «centroïde» qui représente l'ensemble de l'espace des solutions sous-optimales. La notion de «forme» (*Shape*) peut également être utilisée pour parcourir la totalité de l'espace des solutions sous-optimales dans un intervalle d'énergie choisi et calculer la probabilité de chaque structure d'appartenir à cette «forme». Ce type d'approche est notamment utilisé par les outils **RNAshapes** [Steffen06], **RNAcast** [Reeder05], **RNAlishapes** [Voss06], [Lorenz08], ...

3.2.3.2 Les approches utilisant l'intelligence artificielle

Certains algorithmes de prédiction de structures secondaires utilisent un échantillonage statistique de structures connues pour construire un modèle qui pourra alors être utilisé pour prédire la structure d'une séquence donnée. L'algorithme **CONTRAFold** [Do06] utilise ce type d'approche. Sa force repose dans le fait que l'algorithme a été entraîné sur une vaste sélection de structures de la base **Rfam** [Griffiths-Jones03, Griffiths-Jones05b] et contient également de l'information thermodynamique fournie par un modèle reposant sur les grammaires stochastiques non contextuelles. Ce type de grammaires est également utilisé par **CONSAN** [Dowell06] et **PFOLD** [Knudsen03].

De nombreuses approches reposent sur des algorithmes génétiques. Ainsi, le programme **STAR** [van Batenburg95, Gultyaev95] propose une implémentation d'algorithme génétique destiné à être utilisé sur des ordinateurs personnels et se basant sur un petit nombre de structures coévoluant. A l'inverse, il existe aussi des algorithmes génétiques pour la prédiction de structures secondaires destinés à être massivement parallèles. **MPGAfold** [Shapiro01] permet de faire évoluer en parallèle une population de milliers

de structures d'ARN. Cette population pourra être visualisée avec un outil adapté [Shapiro06].

Algorithme génétique

Un algorithme génétique est une méthode visant à trouver une solution sous optimale à un problème d'optimisation. Il s'inspire de la théorie de l'évolution, selon laquelle plus un individu est adapté à son milieu, plus il a de chances de survivre, de se reproduire et donc de transmettre, *via* ses gènes, une partie de ses caractéristiques personnelles. Dans le cadre des algorithmes génétiques, les individus sont des solutions potentielles à un problème d'optimisation et leurs caractéristiques, déterminées aléatoirement, sont codées sur des chromosomes. L'algorithme se déroule de la façon suivante :

1. Choix d'une (très grande) population initiale de solutions
2. Évaluation de l'adaptation (qualité) de chaque individu
3. pour un très grand nombre de générations
   (a) Choix des individus les plus adaptés comme reproducteurs
   (b) Croisement (reproduction sexuée avec crossover et mutations)
   (c) Évaluation de l'adaptation de chaque descendant
   (d) Mort des individus les moins adaptés (parmi les parents et les enfants)

L'introduction de hasard dans la sélection des reproducteurs et des individus qui meurent, couplée aux mutations lors de la reproduction, empêche l'algorithme de converger vers un maximum local.

L'ARN n'est pas une molécule statique et passe par différents états, actifs ou non, durant sa vie dans la cellule. C'est pourquoi certains algorithmes reposent sur une approche cinétique. Le programme **`KINEfold`** [Xayaphoummine03, Xayaphoummine05] utilise une méthode de Monte Carlo pour produire des simulations probabilistes de la cinétique du repliement pour des séquences allant jusqu'à 300-400 nucléotides.

3.2.3.3 La prédiction de structures avec pseudonœuds

Algorithmiquement, la prédiction d'une structure secondaire avec pseudonœuds est un problème qualifié de *NP complet*. Un problème *NP complet* repond à deux propriétés :

- on peut vérifié la validité d'une solution à ce problème en un temps polynomial
- si un probèle *NP complet* peut-être résolue de façon *rapide* (temps polynomial), alors tous les probèles *NP complets* peuvent l'être

A l'heure actuelle, il n'existe aucune solution *rapide* pour résoudre les problèmes *NP complet*. Cela signifie que le programme qui cherche la solution optimale à ce problème est inutilisable (voir le tableau **3.1**). Une structure secondaire d'ARN peut être schématisée mathématiquement par un arbre. Quand cette structure contient des pseudonœuds, une schématisation par un graphe est nécessaire, ce qui augmente considérablement la complexité des algorithmes. Ainsi, un algorithme de programmation dynamique prédisant une structure sans pseudonœud a une complexité moyenne en O(n^3). Un programme de prédiction de structure avec pseudonœuds comme **`Pseudoknots`** [Rivas99] (capable de prédire uniquement une catégorie particulière de pseudonœuds) a une complexité en O(n^6). **`NUPACK`** [Dirks03] et **`pknotsRG`** [Reeder07b] ont une complexité respective de O(n^5) et O(n^4).

Pour qu'un programme de prédiction de structure secondaire avec pseudonœuds puisse s'exécuter dans une temps raisonnable, celui-ci peut se baser sur une heuristique plutôt que sur un algorithme. En plus, à l'aide de certaines simplifications du modèle énergétique sous-jacent ou de la topologie des pseudonœ]uds, le programme parcourra un espace de recherche plus petit. Les outils **`STAR`** [Gultyaev95], **`ILM`** [Ruan04] et **`HotKnots`** [Ren05] utilisent ce type d'implémentation.

L'algorithme proposé par [Liu06] permet de prédire des structures avec pseudonœuds. Des pondérations dynamiques sur la longueur des tiges sont introduites. Une procédure récursive prédit des structures secondaires en cherchant à chaque étape les structures dont la somme des pondérations des tiges est maximale.

Sur la base de certains critères, **KnotSeeker** [Sperschneider08] trouve des fragments de séquences d'ARN pouvant présenter des pseudonœuds. Ces quelques candidats sont ensuite repliés par un algorithme de programmation dynamique pour vérifier si ils peuvent effectivement former des pseudonœuds énergétiquement stables. Les programmes **KINEfold**, **MPGAfold** et **MC-Fold** sont également capables de prédire ce type de structures.
Par ailleurs, **Pseudobase** [van Batenburg00, van Batenburg01] est une base de données contenant des pseudonœuds déterminés expérimentalement et informatiquement.
Enfin, il est intéressant de noter que de nombreuses méthodes cherchent à éliminer les pseudonœuds afin de comparer des structures d'ARN entre elles [Smit08].

3.2.3.4 Les autres méthodes de prédiction de structures secondaires

[Horesh07] produit une structure secondaire, et une série de structures secondaires énergétiquement sous-optimales pour chaque séquence d'un jeu de molécules d'ARN non alignées. Un grand nombre de structures possibles pour chaque séquence est calculé par RNAsubopt [Wuchty99] du **RNA Vienna Package**. Les structures correspondent aux sommets d'un graphe, et sont reliés par des arêtes dont la pondération représente la distance entre ces structures. **RNAspa** cherche alors le plus court chemin passant une fois et une seule par une structure de chaque séquence.
caRNAc [Touzet04, Touzet07] utilise des informations de thermodynamique, de phylogénie et de conservation de séquences pour déduire les éléments de structure secondaire conservés au sein d'une famille d'ARN. **caRNAc** prédit les hélices pour chaque séquence d'ARN, en déduit la structure commune pour chaque paire de séquences et enfin infère une structure globale pour la famille d'ARN.
BayesFold [Knight04] utilise le théorème de Bayes pour intégrer différentes données telles que les paramètres thermodynamiques, la covariation et d'éventuelles données de sondage chimique. Ces données sont ensuite utilisées pour inférer une structure

secondaire.

RDFolder [Ying04] prédit une structure secondaire pour une séquence donnée en utilisant deux méthodes. Pour les séquences courtes, une méthode Monte Carlo produit de nombreuses structures secondaires par empilement aléatoire de paires de bases, la structure la plus fréquemment générée est sélectionnée. Pour les séquences longues, 1) une méthode de distribution des hélices produit de nombreuses structures secondaires à l'aide de la méthode précédente. 2) L'hélice la plus fréquente est sélectionnée, puis les hélices incompatibles avec celle-ci sont supprimées et 1) et 2) sont répétés jusqu'à ce qu'aucune nouvelle hélice ne soit prédite.

SEED [Anwar06] utilise des tableaux de suffixes pour établir des motifs structuraux de l'une des séquences d'entrée, choisie comme graine (*seed*). Ces motifs sont ensuite validés ou rejetés en fonction de leur conservation dans les autres séquences de l'alignement. Les motifs conservés permettent de définir une structure secondaire consensus au jeu de séquences.

La revue suivante détaille ces différentes approches [Gardner04]. La page [ssprediction] constitue également un bon rappel des différentes approches.

3.3 La prédiction de structures tertiaires

3.3.1 La prédiction de structures tertiaires à partir d'une structure secondaire

L'outil **MC-Sym** [Parisien08] (détaillé dans le chapitre **6**) permet de produire un repliement tridimensionnel d'ARN en se basant sur des contraintes structurales contenues dans une structure secondaire, étendue à la représentation des paires de bases non canoniques. L'algorithme sous-jacent extrait de cette structure secondaire étendue, un réseau d'interactions (paires et empilements de bases). Ce réseau est ensuite décomposé en cycles d'interactions. Pour chacun de ces cycles, un équivalent

tridimensionnel est choisi dans une banque (obtenue par «découpage» des structures résolues présentes dans la **Protein Data Bank** [Berman00]). Ces différents équivalents sont ensuite assemblés pour composer la structure tertiaire globale. L'outil **RNA2D3D** [Martinez08] (détaillé dans le chapitre **6**) propose de générer un modèle de structure tertiaire éditable à partir d'une structure secondaire. Ce programme produit un dessin de structure secondaire avec des contraintes particulières de distance entre les nucléotides, afin de respecter la règles de la chimie. Ce dessin est ensuite étendu pour incorporer les atomes des nucléotides. Les hélices, planes à l'origine, subissent alors une torsion autour de leur axe pour adopter la conformation de type A. Les simples brins sont fixés de façon rigide en fin d'hélice.
Le programme **ERNA-3D**, utilisé notamment dans l'étude d'hybrides entre ARN de transfert et ARN messagers [Burks05], permet de générer automatiquement un premier jet de modèle de structure tertiaire sous la forme d'hélices régulières, reliées par des simple brins. L'utilisateur peut alors éditer la structure en déplaçant des hélices et des nucléotides. Les nucléotides sont reliés de façon élastique, et lorsqu'un nucléotide est déplacé, les angles de torsion entre celui-ci et son voisin sont modifiés en temps réel, assurant que les différentes parties du modèle restent connectées tout en garantissant un respect des règles stéréochimiques (distance, angle planaire et angle dihèdre). Il est également possible de visualiser des cartes de densité électronique. Toutefois, ce programme est payant ($1.500 pour une licence illimitée) et ne fonctionne que sur des station de travail SGI avec Irix 6.5

3.3.2 La prédiction de structures tertiaires à partir d'une séquence

Le logiciel **FARNA** [Das07] (détaillé dans le chapitre **6**) cherche à prédire, à partir d'une séquence, la structure tertiaire d'énergie minimale, en se reposant sur une base de connaissances de fonctions d'énergie et de conformation du squelette ribose phosphate. Cette méthode s'inspire de la méthode **Rosetta** [Simons97] qui prédit des structures

de protéines à basse résolution. Le coupage des outils **MC-Fold** et **MC-Sym** permet d'obtenir le même type de résultats.

3.4 L'annotation et la comparaison de structures d'ARN

Lorsque l'on possède la structure tridimensionnelle d'un ARN, il est intéressant de la comparer aux structures d'ARN homologues afin d'identifier, par exemple, des domaines conservés, insérés, altérés ou supprimés. Toutefois, une comparaison atome par atome ne semble pas être une approche raisonnable (car techniquement non réalisable sur des molécules de grandes tailles). C'est pourquoi, la structure de l'ARN devra être simplifiée et schématisée, afin de ne porter que l'information nécessaire et suffisante à une telle comparaison. Ce processus est appelé «*annotation de structure tertiaire*».

3.4.1 L'annotation de structures tertiaires

L'étape d'annotation de structure tertiaire consiste à prendre toutes les données chimiques disponibles dans une structure tertiaire et d'en produire une vue simplifiée, mais comprenant toutefois toutes les informations nécessaires à la bonne compréhension de la structure. De manière générale, on va produire une structure secondaire à laquelle seront ajoutées les informations d'interactions tertiaires qui comprendront notamment :

- les paires de bases canoniques
- les paires de bases non canoniques
- l'empilement des bases

Plusieurs approches indépendantes, mais néanmoins analogues ont été développées, on peut noter en particulier les logiciels **MC-Annotate** [Lemieux02], **RNAview**

[Yang03] et **FR3D** [Sarver08]. L'algorithme **RNAview** est détaillé en **5.4.8.1**, illustrant le principe de l'annotation de structures tertiaires.

3.4.2 La comparaison de structures & la recherche de motifs

Comme nous l'avons vu dans le chapitre précédent, les motifs structuraux sont des éléments primordiaux des molécules d'ARN, car ce sont ces motifs qui donnent à l'ARN le repliement qui le rend fonctionnel, en permettant de rapprocher des éléments distants (Kink-Turn, C-Loop) ou en créant des sites de fixation (GNRA, Sarcin-Ricin). Il est donc vital d'étendre notre connaissance en découvrant de nouveaux motifs et en les identifiant dans de nouvelles molécule d'ARN. La revue [Leontis06b] présente la notion de motif structural ainsi qu'une série de programme permettant leur construction et leur identification.

***RMSD** : **R**oot **M**ean **S**quare **D**eviation*

La RMSD entre deux jeux de points (en trois dimensions) est la racine que la distance moyenne point-à-point entre les deux jeux, lorsque leur superposition est maximale. Cela correspond à la distance moyenne entre les deux jeux de points.

Soit θ_1 et θ_2 deux jeux de n points dans $\mathbf{R}^3$ tels que :

$$\theta_1 = \begin{pmatrix} x_{1,1} & y_{1,1} & z_{1,1} \\ x_{1,2} & y_{1,2} & z_{1,2} \\ \vdots & \vdots & \vdots \\ x_{1,n} & y_{1,n} & z_{1,n} \end{pmatrix} \text{ et } \theta_2 = \begin{pmatrix} x_{2,1} & y_{2,1} & z_{2,1} \\ x_{2,2} & y_{2,2} & z_{2,2} \\ \vdots & \vdots & \vdots \\ x_{2,n} & y_{2,n} & z_{2,n} \end{pmatrix}$$

La RMSD entre θ_1 et θ_2 s'obtient par :

$$RMSD(\theta_1, \theta_2) = \sqrt{\frac{\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2 + (y_{1,i} - y_{2,i})^2 + (z_{1,i} - z_{2,i})^2}{n}}$$

La recherche de motifs peut avoir lieu au sein de structures tridimensionnelles. L'outil **FR3D** [Sarver08] introduit la notion d'*écart géométrique* entre deux jeux de

n nucléotides. Quand ces deux jeux sont superposés de façon à avoir une RMSD minimale, on calcul la différence d'orientation de chaque couple de nucléotides homologues. L'écart géométrique est la racine de la somme de ces n différences d'orientations combinée à la RMSD obtenue par la superposition. L'utilisateur décrit le motif structural en sélectionnant les nucléotides qui le forment dans une structure tridimensionnelle. **FR3D** parcours alors l'ensemble des structures à sa disposition pour retrouver toutes les occurrences de ce motif. C'est à dire toutes les sélections possibles de nucléotides qui ont un écart géométrique inférieur à un seuil donné par l'utilisateur.

NASSAM [Harrison03] propose une simplification de la structure tridimensionnelle d'un ARN en symbolisant la base de chaque nucléotides par deux vecteurs. Cette représentation simplifiée permet de trouver des motifs récurrents dans des banques de données de structures tridimensionnelles d'ARN. Le motif à chercher et la structure étudiée sont convertis en graphes, et la recherche de candidats dans la structure se fait à l'aide de l'algorithme d'isomorphisme de sous-graphe de Ullmann.

Le programme **ARTS** [Dror05, Dror06] prend en compte les coordonnées des phosphates et des paires de bases de deux structures tridimensionnelles et en produit un alignement. Cette alignement n'est pas représenté nucléotide par nucléotide mais consiste en la «meilleure» superposition des deux structures, mettant en valeur les regions conservées. Les paires de bases homologues sont listées et l'utilisateur peut explorer une série d'alignement sous-optimaux (la RMSD étant le score classant les alignements). Le parcours de ces solutions sous-optimales est nécessaire car les structures sont alignées de façon rigide, et donc une forte superposition dans une region de la structure ne permettra pas de mettre en évidence une grande conservation dans une region distante.

Une approche a été proposée dans [Huang05] pour superposer et partitionner l'ensemble des boucles apicales disponibles dans la banque SCOR. Des familles telles que les boucles GNRA et UNCG ont pu être mises en valeurs par ce partitionnement.

Dans [Lemieux06] la structure tertiaire de la grande sous unité du ribosome de *H. marismortui* a été décomposée en un graphe. Ce graphe est formé des cycles

minimaux d'interactions entre les bases (empilements et appariements canoniques ou non canoniques). Ces différents cycles ont été classés en fonction de leur nombre d'occurrences, et l'on constate sans surprise que le cycle d'interaction le plus fréquent entre quatre bases est l'empilement de deux paires de bases canoniques. Le second cycle le plus fréquent est le motif GNRA, dont une étude est présentée dans cet article. Le programme **MC-Cycle** [St-Onge07] repose lui aussi sur cette décomposition de structure en cycles. L'utilisateur fournit un motif structural, en le décrivant à l'aide cycle d'interactions, et **MC-Cycle** est alors capable de générer toutes les séquences compatibles avec ce motif. Dans ces approches, il apparaît que le squelette sucre-phosphate ne joue qu'un rôle mineur dans l'établissement des paires de bases. En effet, des bases appartenant à des nombres différents de brins pourront former un même motif. Toutefois, des programmes sont également spécialisés dans la recherche de motifs du squelette sucre-phosphate.

Le squelette ribose phosphate est articulé entre chaque nucléotide par six angles de torsion. La quantité importante d'angles le long d'un brin rend vite les calculs conformationnels très coûteux. Des approches visent donc à simplifier la représentation de ce squelette afin de rendre les calculs possibles. **Primos** [Duarte03] et **COMPADRES** [Wadley04] proposent d'utiliser deux pseudo angles de torsion η et θ défini par les atomes C4' et P. Cette représentation simplifiée est alors utilisée pour chercher des motifs récurrents de squelette dans des structures d'ARN.

La recherche de motifs peut également le faire au niveau de la séquence et de la structure secondaire. **RNAMOT** permet de trouver les séquences compatibles avec un motif décrit par des éléments de séquence et de structure secondaire fournis en entrée. **RNABOB** est une implémentation alternative de **RNAMOT** dont l'algorithme repose sur un automate non déterministe à états finis.

RNAMotif [Macke01] et **PatSearch** [Grillo03] permettent de définir un motif de structure secondaire en fournissant une liste d'interactions, et cherchent dans des banques génomiques des séquences compatibles avec ce motif.

Locomotif [Reeder07a] propose à l'utilisateur de fournir le motif de structure

secondaire qu'il souhaite rechercher à l'aide d'un éditeur. L'utilisateur décrit le motif à l'aide d'éléments structuraux tels que «hélice», «bulle interne», «boucle apicale»,... Ces éléments peuvent avoir une longueur et/ou une séquence déterminée.
La «forme» [Steffen06] du motif est ensuite utilisée pour recherche des cibles compatibles dans un jeu de séquences.

`RNAProfile` [Pavesi04] cherche dans un jeu de séquences non alignées, les regions de chaque séquence présentant un nombre d'épingles à cheveux défini par l'utilisateur. L'algorithme regroupe alors les regions trouvées par similarité, permettant ainsi de mettre en évidence des motifs de structure secondaire conservés entre plusieurs regions.

`CMfinder` [Yao06] recherche dans une jeu de séquences non alignées les éléments de structure secondaire les plus conservés, dont le nombre et la longueur des hélices sont définis par l'utilisateur.

`RNAmine` [Hamada06] cherche des motifs récurrents d'hélices dans des séquences non alignées, en extrayant de l'information structurale des séquences. Cet outil calcule les probabilités d'appariement de chaque base, mais ne produit pas de structure secondaire complète pour chaque séquence.

Une étude des différentes stratégies pour mesurer la conversation de la structure secondaire dans un alignement est présentée dans [Gruber08]. Plusieurs approches pour détecter des structures secondaires conservées au sein d'alignements multiples ont été évaluées. Il apparaît que les méthodes reposant sur l'indice de conservation de structure (SCI), telle que **`RNAz`** [Washietl05] (voir 3.6), sont globalement les plus efficaces.

`LGSFAligner` [Jansson06] propose de trouver des motifs communs à deux structures secondaires en représentant ces structures par des forêts qui seront ensuite alignées par le programme de manière analogue à [Höchsmann03].

`RNACluster` [Liu08] propose une méthode pour partitionner un jeu de structures secondaires. L'utilisateur a le choix entre cinq algorithmes différents pour mesurer la distance entre les structures.

`ERPIN` [Lambert04] permet de chercher des séquences compatibles avec un motif

structural qui est automatiquement déduit d'un alignement de séquences donné en entrée du programme.

3.5 Les bases de données relatives à l'ARN

Les informations biologiques publiquement disponibles actuellement sont très nombreuses, de natures variées, et réparties dans de nombreuses bases de données hétérogènes. Outre un nombre toujours grandissant de génomes, la communauté scientifique a aujourd'hui accès à de nombreuses bases de données dédiées uniquement ou en partie à l'ARN. Ces banques peuvent être axées sur une famille moléculaire comme la RNase P avec la **Ribonuclease P Database** [Brown99]. La **Genomic tRNA Database** fournie la liste des ARN de transfert trouvés dans les génomes par le programme **tRNAscan-SE** [Lowe97]. La banque **Mamit-tRNA** [Pütz07] fournit une liste de séquences et structures secondaires d'ARN de transfert mitochondriaux.
Il existe également de nombreuses bases de données dédiées aux microARN comme **Argonaute** [Shahi06]. La base **miRDB** [Wang08] se focalise plus particulièrement sur les cibles potentielles de ces microARN, alors que **miRNApath** [Chiromatzo07] est dédiées aux micro ARN, à leurs cibles et aux voies métaboliques qu'ils affectent. D'autres banques se consacrent à une famille d'organismes, par exemple **ASRP** [Gustafson05, Backman08] est dédiée aux ARN non codants de petites tailles des plantes. **RNAdb** [Pang05, Pang07] contient des séquences d'ARN non codants de mammifères, accompagnées de nombreuses informations (telles que les pathologies ou processus développementaux affectés par ces ARN).
Il existe également des banques de données plus généralistes. Ainsi, **NONCODE** [Liu05a, He08] et **ncRNAdb** [Szymanski07] se consacrent à répertorier tous types d'ARN non codants et fournissent diverses informations utiles les concernant. **fRNAdb** [Kin07] contient une collection de transcrits non codants pouvant être annotés, et ayant subi un certain nombre d'analyses comme la recherche de motifs de structure secondaire.

Plus d'informations structurales sont disponibles dans la banque **Rfam** [Griffiths-Jones03, He08], qui est une collection d'alignements multiples de séquences d'ARN. Les ARN sont répartis par familles et un modèle de covariance est disponible pour chaque famille. La base **SCOR** [Klosterman02, Tamura04], quant à elle, est une collection de motifs structuraux d'ARN résolus en 3D par diffraction de rayons X ou résonance magnétique nucléaire. Elle propose une grande quantité de boucles internes et externes ainsi que des motifs plus complexes comme le Kink Turn. De façon analogue, **RNA FRABASE** [Popenda08] est une base de données permettant de trouver des fragments de structures tridimensionnelles d'ARN en fonction d'une séquence ou structure secondaire. La **NCIR** (Non-Canonical Nase Pair Database)[Nagaswamy02, ncir] a pour objectif de fournir un accès rapide à toutes les structures dans lesquelles des paires de bases particulièrement rares ont été observées et de décrire les propriétés associées à chaque paire de bases non-canonique. Enfin, la **Nucleic Acid Database** (**NDB**) [Berman02, Berman03] contient des structures tridimensionnelles de molécules d'ARN, obtenues par diffraction de rayons X ou par résonance magnétique nucléaire. Elle est l'équivalent pour les ARN de la **Protein Data Bank** (PDB) [Berman00].

3.6 La recherche de molécules d'ARN dans les génomes

Depuis l'existence des banques de données génomiques, la recherche de molécules homologues est une tâche récurrente. Initialement, ces recherches se faisaient par pure homologie de séquences, à l'aide d'outils tels que **BLAST** [Altschul90]. L'étude structurale des ARN a permis de rendre ce travail plus efficace en apportant des éléments de comparaison supplémentaires entre les molécules. Ainsi, il est possible de rechercher une molécule compatible avec certains éléments de structure secondaire (**RNAMST** [Chang06]) ou certains motifs structuraux (**MilPat** [Thebault06]). **HomoStRscan** [Le04] permet de chercher, dans des génomes complets, les homologues d'une molécule

d'ARN, dont l'utilisateur a fourni la séquence et la structure secondaire. Pour ce faire, le programme fait glisser une fenêtre de la taille de la séquence le long du génome, prédit la structure secondaire de la séquence dans la fenêtre et mesure sa similarité avec la structure fournie par l'utilisateur. Le programme a notamment été utilisé avec succès pour trouver l'ARN ribosomal 5S dans trois génomes bactériens. Le couplage des outils **RAVENNA** [Weinberg06] et **RSEARCH** [Klein03] permet quant à lui de chercher des homologues en utilisant un alignement comme référence.

L'aspect de l'ARN comme régulateur de l'expression génétique jouant un rôle important dans de nombreuses pathologies (virus, cancer, ...), certains outils sont dédiés à trouver des candidats de microARN et/ou de leurs cibles potentielles (**miRNAlign** [Wang05], **microHARVESTER** [Dezulian06], **RNAmicro** [Hertel06]).

D'autres outils cherchent à identifier des ARN non codants sans utiliser de séquence, de structure ou de familles comme référence et procèdent donc à une recherche sans *a priori*. **RNAz** [Washietl05] est un programme capable de parcours un alignement de génomes pour chercher des candidats d'ARN non codants. L'approche se basent sur une combinaison d'un score de stabilité thermodynamique et un indice de conservation de structure (SCI) obtenu par une mesure de la covariation entre les séquences. Ce programme est efficace sur des alignements de génomes de grandes tailles. D'autres programmes permettent également de chercher des ARN non codants dans des génomes grâce à la mise en évidence de structures secondaires conservées dans les alignements de ces génomes. L'outil **QRNA** [Rivas01a] est l'un des plus utilisés et a permis d'identifier des ARN non codants dans des génomes de bactéries [Rivas01b] et de levures [McCutcheon03]. Toutefois, il est trop coûteux pour être utilisé sur des génomes de grandes tailles et est limité à des alignement de deux génomes. **ddbRNA** [di Bernardo03] présente ces mêmes faiblesses. **MSARI** [Coventry04] quant à lui, est capable de faire une recherche dans des alignements d'une dizaine de génomes, ce qui apporte une grande richesse d'information et rend le programme efficace. Cependant, on ne dispose à l'heure actuelle que de peu d'alignements de cette taille.

3.7 Coévolution

Comme nous l'avons vu, le bon fonctionnement d'un ARN constitue une importante pression de sélection. En effet, un ARN avec une mutation invalidant sa fonction sera rapidement éliminée par l'évolution. Ainsi, lorsque l'on observe une mutation en un point de la molécule, on observe souvent une mutation compensatoire en un autre point. Cette co-variation assure la conservation de la fonction de l'ARN. Toutefois, un ARN n'est pas seul dans la cellule, il est entouré d'autres molécules qui évoluent également. Ainsi, on observe parfois des apparitions, disparitions ou mutations de domaines structuraux complets, qui sont compensés par d'autres mutations au sein des molécules partenaires de cet ARN. Ces mutations sont donc difficilement explicables en ne regardant qu'une molécule et il est important de pouvoir comparer l'évolution de plusieurs molécules (ARN ou protéines). Cette comparaison de systèmes entiers représente un travail considérable et sera vraisemblablement l'un des grands chantiers de la recherche biologique et bioinformatique.

3.8 Visualiser et manipuler les données bioinformatiques de l'ARN

Les diverses informations relatives à l'ARN peuvent être très complexes, notamment pour des molécules de grandes tailles. Il est donc nécessaire d'avoir des outils adaptés à la visualisation de telles informations. Il existe de nombreux visualiseurs de structures tridimensionnelles, parmi les plus utilisées, on trouve **PyMOL** [pymol], **VMD** [Humphrey96] et **Chimera** [Huang96]. Il existe également des outils permettant de visualiser des structures secondaires d'ARN comme **RNAMLView** [Yang03] **S2S** [Jossinet05] ou **RnaViz** [De Rijk97, De Rijk03]. Certains outils sont adaptés aux alignements structuraux, c'est le cas notamment de **S2S**, **SARSE** [Andersen07]

et **`Colorstock`** [Bendaña08]. **`SQUINT`** [Goode07] est un éditeur d'alignements multiples d'acides nucléotides ou de protéines, qui propose également un algorithme de programmation dynamique permettant d'obtenir un premier alignement qui sera ensuite affiné par l'utilisateur. Ce programme offre la possibilité intéressante de pouvoir visualiser simultanément deux positions distantes de l'alignement et de recalculer en temps réel le score de l'alignement, mais n'offre malheureusement aucune information structurale.

Au cours des dernières années, plusieurs revues portant sur la bioinformatique des ARN en générale, ou sur l'une de ses spécialités, ont été publiées. Les articles suivants sont d'un intérêt particulier [Leontis06b, Eddy06, Shapiro07, Kochiwa06]. Le site [ncrna] est également une bonne source d'informations, et répertorie notamment un grand nombre d'outils bioinformatiques non publiés. L'article [Jossinet07] qui clôture ce chapitre constitue une revue récente sur la bioinformatique des ARN.

3.9 Les objectifs de cette thèse

Comme nous nous intéressons à la structure des ARN, nous avons un réel besoin d'avoir toujours plus de structures tridimensionnelles. Toutefois, les structures produites par des approches de types biophysiques (cristallographie, résonance magnétique, ...) sont considérablement moins nombreuses que les séquences pour ces ARN. Nous avons vu que des algorithmes tentent de combler cet écart. Malheureusement, ils s'avèrent très coûteux en ressources et peu efficaces lorsque la taille des molécules devient très importante.
Nous avons donc décidé de développer un outil appelé **`Assemble`** permettant de construire des structures tridimensionnelles d'ARN de grande taille par modélisation moléculaire. Cette construction sera réalisée entièrement par l'utilisateur, ce qui garantit une plus grande fiabilité que pour un travail réalisé par un algorithme sans supervision. De nombreux automatismes permettent de réduire le temps de construction en proposant à l'utilisateur de faire appel à des algorithmes à différentes étapes de la modélisation. Cet outil est présenté dans le chapitre **7**.

Par ailleurs, nous avons vu que l'ARN présente différents niveaux d'analyse et qu'il existe de nombreux algorithmes et outils graphiques permettant l'étude de chacun de ces niveaux. Ces outils constituent une vaste source d'informations pour la compréhension des ARN et leur utilisation représente un atout majeur pour la construction de modèles moléculaires.
Toutefois, ces différents outils, développés par des équipes indépendantes n'utilisent pas systématiquement les mêmes standards et formats de fichiers, il est donc difficile de les interconnecter. Il est donc nécessaire d'unifier les données qu'ils exploitent au moyen d'un formalisme commun. Pour cela, **`Assemble`** et les algorithmes qu'il exploite reposent sur la plateforme **`P.A.R.A.DIS.E`** présentée en chapitre **5**.

P.A.R.A.DIS.E est une plateforme d'analyse des annotations d'ARN composée :

- d'interfaces graphiques destinée à la représentation visuelle de données d'ARN
- d'un modèle informatique des concepts liés à l'ARN
- d'une couche de communication permettant l'échange d'informations entre les interfaces graphiques et les algorithmes exploités par **P.A.R.A.DIS.E**

Dans une dernière partie, nous avons éprouvé nos divers outils bioinformatiques dans des projets de modélisation de structures tridimensionnelles de grande envergure. Nous avons notamment modélisé des structures de taille moyenne (250 nucléotides) en nous basant sur une structure secondaire obtenue à l'aide d'alignements de séquence et de données de biologie expérimentale. Nous avons également travaillé sur la modélisation d'une structure de grande taille (la grande sous unité du ribosome de *S. cerevisiae*) en utilisant des cartes de densité issues de la cryo-microscopie électronique.

Chapitre 4

Article 1

RNA structure : bioinformatic analysis

F. Jossinet, TE. Ludwig & E. Westhof

Current Opinion in Microbiology (2007) **10** :279-285

Résumé

L'étendue des fonctions attribuées aux molécules d'ARN a considérablement augmentée durant ces dernières années. Par conséquent, l'analyse et la comparaison des séquences d'ARN sont devenues des tâches récurrentes en biologie moléculaire. Comme la fonction biologique d'un ARN est exprimée plus par son repliement que par sa séquence, des outils informatiques originaux, adaptés aux multiples facettes des ARN doivent être développés. De tels outils, récemment publiés, permettent à un utilisateur de résoudre des problèmes classiques liés à la recherche sur les ARN : construire des alignements «structuraux» multiples, déduire des structures complètes et des motifs structuraux de ces alignements, ou rechercher des homologues structuraux dans des banques de données génomiques.

RNA structure: bioinformatic analysis

Fabrice JOSSINET, Thomas E. LUDWIG and Éric WESTHOF

Current Opinion in microbiology, 2007, Vol. 10, Numéro 3, Pages 279–285

Deuxième partie

P.A.R.A.DIS.E : une plateforme d'analyse des annotations d'ARN

Chapitre 5

L'infrastructure P.A.R.A.DIS.E

5.1 Introduction

5.1.1 L'intégration des programmes et des données bioinformatiques

L'étude des molécules d'ARN est une problématique complexe, car la compréhension de leur fonction biologique nécessite la mise en place et l'interconnexion de plusieurs niveaux d'analyse : séquence, structure secondaire et tertiaire, données de sondage chimique, alignement de séquences, interactions avec d'autres biomolécules, ... qu'il est nécessaire de recouper. Comme nous l'avons vu au chapitre **3**, pour chacun de ces niveaux, on trouve un nombre important et toujours croissant d'outils et de banques de données dont il faudra exploiter les résultats. La confrontation de ces résultats est toutefois une tâche difficile, tous ces outils ayant été développés par des laboratoires différents, utilisant des concepts et des formats de fichiers hétérogènes. Il existe donc un besoin d'intégration des outils existants, qui n'est pas propre à l'étude des ARN mais se place dans la problématique générale d'intégration

des données biologiques [Stein02].
Les progrès dans des domaines complexes comme la biologie systémique sont entravés par le faible nombre d'infrastructures logicielles adaptées à leur étude. Pour accélérer le développement de telles infrastructures, qui sont composées d'éléments complexes, il est nécessaire d'intégrer des outils déjà existants. Pour cette intégration, il y a un besoin de standardisation au niveau des techniques, et d'unification au niveau des concepts. Au delà de la standardisation, ces outils intégrés doivent pouvoir être enchaînés, de manière personnalisée, pour former de réels protocoles expérimentaux bioinformatiques [Swertz07].
L'intégration de multiples ressources bioinformatiques peut se résumer à un problème de communication. Cette communication doit pouvoir s'effectuer d'un outil à l'autre, mais également avec le chercheur qui va valider, éditer ou rejeter les informations biologiques fournies par ces outils et qui souhaitera également créer de nouvelles informations. La mise en place d'infrastructures permettant cette communication nécessitera donc :

- des interfaces graphiques permettant la visualisation, l'édition et la création d'informations biologiques
- un langage permettant la modélisation de concepts et d'informations biologiques échangés par les outils
- un médium de la communication permettant cet échange d'informations

Au cours des dernières années, de nombreuses initiatives ont été prises dans ce sens à différents niveaux.

Les interfaces graphiques

Afin de bien comprendre les résultats des algorithmes, il est nécessaire de disposer d'interfaces graphiques adaptées, permettant une représentation visuelle parlante de ces informations. Ces interfaces devront également permettre au chercheur d'éditer les informations affichées, afin de les affiner et de créer de nouvelles informations *de novo*

pour refléter des données qui ne sont pas issues d'analyses algorithmiques.
Le fait de pouvoir interconnecter ces interfaces ou de les utiliser pour contrôler directement les algorithmes sont des fonctionnalités intéressantes qui nécessitent une unification de la modélisation des données biologiques.

Les librairies de concepts biologiques

De nombreuses infrastructures, ayant vu le jour dans les dernières années, proposent une implémentation de concepts biologiques fondamentaux au sein d'un modèle informatique, accessible le plus souvent au travers d'une interface de programmation (API). La mise en place des infrastructures nécessite auparavant l'élaboration d'un vocabulaire décrivant les concepts existants, leurs propriétés, les relations entre ces concepts et permettant d'identifier de façon unique les différents *objets* les implémentant. Ce type d'initiatives souligne le besoin d'ontologies appliquées au données biologiques. Le rôle des ces ontologies est de fournir une représentation unifiée des concepts d'un domaine particulier, par la définition d'un vocabulaire commun, contrôlé et structuré permettant le partage d'informations entre hommes et machines [Gruber93]. Une ontologie contient donc des définissions, compréhensibles par l'homme et interprétables par la machine, des entités d'un domaine, de leurs propriétés, des relations qui les relient et des contraintes régulant ces relations [Smith05]. De nombreuses ontologies existent actuellement dans les sciences de la vie, parmi lesquelles on trouve la **`Gene Ontology`** [Ashburner00] qui visent à décrire les gènes et leurs produits en développant un vocabulaire reparti sur trois ontologies : *fonction moléculaire*, *processus biologique* et *composant cellulaire* et en l'utilisant pour annoter les gènes. La **`Gene Ontology`** fait partie d'un effort de classification plus vaste : les **`Open Biomedical Ontologies`** (OBO) [Smith07] qui visent l'orthogonalisation et l'interopérabilité des ontologies du domaine biomédical. Le **`RNA Ontology Consortium`** travaille actuellement à la mise en place d'une ontologie dédiées au domaine de l'ARN [Leontis06a].

Grâce à de nombreux outils disponibles actuellement il est possible d'utiliser la description informatique d'une ontologie pour générer automatiquement une API représentant les différents concepts et leurs relations. Ces API peuvent donc servir de fondations communes à de nombreux nouveaux outils et leur permettre d'utiliser la même représentation des données biologiques et donc de faciliter leur communication. Toutefois, pour pouvoir effectivement communiquer, au delà d'un langage commun, ces entités logicielles nécessitent également un vecteur dédié au transport des messages.

Les protocoles de communication

Des initiatives récentes proposent l'implémentation de protocoles de communication permettant l'échange de messages entre des outils bioinformatiques. Parmi les infrastructures proposées, un certain nombre visent à faciliter la création de véritables protocoles expérimentaux bioinformatiques en permettant la mise en place d'enchaînement (*workflow*) entre les différents outils disponibles. Les divers algorithmes reliés par ces protocoles de communication peuvent s'avérer délicats à installer et maintenir, notamment pour des utilisateurs non informaticiens, et leurs exigences en ressources informatiques peuvent être relativement élevées. Les protocoles développés auront donc tout intérêt à permettre le déploiement et l'interconnexion des outils sur plusieurs machines.

Une analyse détaillée des propositions faites dans ces domaines est donnée en **5.3** pour les librairies de concepts et **5.4** pour les protocoles de communication.

5.1.2 P.A.R.A.DIS.E : un exemple d'infrastructure d'intégration des données sur l'ARN

Cette volonté d'intégration s'est traduite au laboratoire par le développement du logiciel **S2S** [Jossinet05] qui permet de visualiser, de façon interconnectée, des informations de séquence, de structure secondaire et de structure tertiaire d'ARN. Pour ce faire, **S2S** propose un visualiseur de structure secondaire, un aligneur de séquences et permet de contrôler le visualiseur 3D **PyMOL**. Ces trois outils sont interconnectés et lorsque l'utilisateur focalise son attention sur une région d'une molécule dans l'un des outils (par exemple en la sélectionnant), les autres outils synchronisent leurs vues. Initialement, ce programme devait répondre à deux besoins :

- l'étude de structures tridimensionnelles de taille conséquente. **S2S** utilise l'algorithme **RNAView** [Yang03] développé par la **NDB** permettant d'annoter la structure tertiaire d'un ARN à l'aide d'une structure secondaire *étendue* (cet algorithme et la notion de structure secondaire étendue sont détaillés dans la partie **5.2.2.1**). Cette structure secondaire est affichée par le visualiseur adaptée et permet à l'utilisateur de naviguer au sein de la structure.
- l'étude d'une famille d'ARN orthologues. L'aligneur de séquences de **S2S** permet d'afficher, au moyen d'un masque structural, le respect de contraintes structurales imposées par une structure de référence telles que le respect ou non de l'isostérie entre des paires de bases alignées.

Il est rapidement apparu que ces deux fonctionnalités pouvaient être grandement améliorées grâce à l'introduction d'algorithmes permettant d'automatiser un certain nombre d'étapes clés :

- la production du dessin de la structure secondaire, pour que celui-ci soit le plus lisible possible
- la construction un premier jet d'alignement structural entre les différentes molécules orthologues

– la prédiction d'une structure secondaire pour les molécules orthologues afin de pouvoir les aligner structuralement contre la molécule de référence

Il serait également intéressant de pouvoir éditer les structures secondaires manipulées dans **S2S** lorsque celles-ci sont produites de façon erronée par les algorithmes, ou lorsqu'elle ne répondent pas à certains critères.
Par ailleurs il est souhaitable d'ajouter à **S2S** la possibilité de construire un modèle de structure tridimensionnelle pour les molécules alignées contre la molécule de référence. Ceci implique l'ajout de nouveaux outils graphiques permettant la création et l'édition de structures tridimensionnelles d'ARN, mais également la présence de nombreux algorithmes optimisant le travail de modélisation.

L'ajout de plusieurs algorithmes est donc nécessaire. Toutefois, ces algorithmes peuvent se montrer lourds à installer et maintenir, et être exigeants en ressources. Chaque utilisateur n'a donc pas forcement envie d'avoir à gérer les algorithmes qu'il utilise. L'utilisation d'une architecture distribuée permettrait d'installer l'ensemble des algorithmes sur un serveur puissant au sein du laboratoire et d'utiliser les interfaces graphiques sur l'ordinateur personnel de chaque utilisateur.
Le développement d'une telle infrastructure nécessite cependant de définir préalablement un langage décrivant les différents concepts utilisés par les outils graphiques et algorithmes ainsi que les relations entre ces concepts. De plus, afin de permettre à ces différentes entités d'échanger des messages construits grâce à ce langage, la mise en place d'un moyen de communication est également nécessaire.

L'ensemble de ces évolutions du logiciel **S2S** entrent dans le cadre du développement d'une plateforme d'analyse des annotations d'ARN nommée **P.A.R.A.DIS.E** (**P**latform to **A**nalyze **R**NA **A**nnotations over a **Dis**tributed **E**nvironment) qui sera donc composée :

– d'interfaces graphiques permettant la visualisation des données de l'ARN

- d'algorithmes permettant la production de données biologiques et l'automatisation des opérations d'édition de l'utilisateur dans les interfaces graphiques
- d'une librairie permettant la modélisation informatique de concepts et d'informations biologiques
- d'une couche de communication permettant l'échange de données biologiques entre les différents modules de la plateforme

S2S et son extension **P.A.R.A.DIS.E** ont été implémentés à l'aide du langage de programmation **Java** [java]. Ce langage présente trois avantages majeurs pour l'implémentation de notre infrastructure :

1. des capacités de modélisation poussées, reposant sur le paradigme de la programmation orientée objet (*POO*), nous permettant de traduire fidèlement les concepts biologiques et leurs relations dans un modèle programmatique
2. la possibilité d'écrire facilement des interfaces graphiques portables, à l'aide de la librairie **Swing**
3. la présence de nombreuses librairies externes permettant de mettre en place des protocoles poussés de communication (voir **5.4**)

L'objectif à long terme de la plateforme **P.A.R.A.DIS.E** est de devenir un système de gestion de l'information du laboratoire (**LIMS : L**aboratory **M**anagement **I**nformation **S**ystem) [Robinson83, McDowall88] focalisé sur les doonées de l'ARN, permettant aux utilisateurs :

1. d'enchaîner des expérimentations bioinformatiques les plus variées possibles, couvrant au maximum le domaine de la bioinformatique des ARN
2. d'échanger facilement leurs données et résultats au sein du laboratoire
3. d'accéder de façon transparente à des bases de données diverses

Dans un premier temps, cette plateforme sera validée en répondant aux attentes du premier point, en permettant d'effectuer les enchaînements d'opérations décrites par la figure **5.1**.

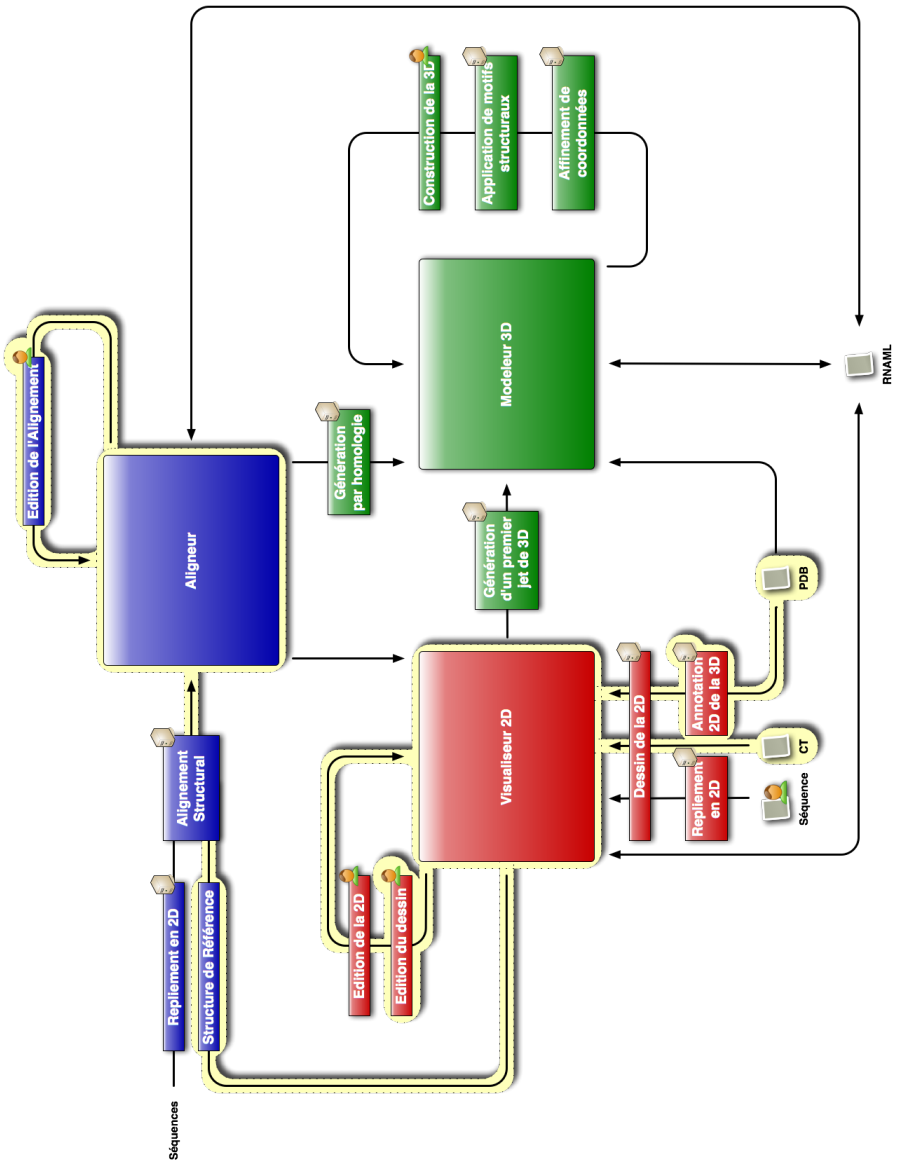


FIG. 5.1 – Les enchaînements d'analyses de **P.A.R.A.DIS.E**

Ce diagramme montre les successions d'analyses possibles avec ***P.A.R.A.DIS.E****. En rouge : la partie structure secondaire, en bleu : la partie alignement structural et en vert la partie structure tertiaire, qui représente l'essentiel de mon travail de thèse. Les analyses et relations sur fond jaune ont été initiées dans* ***S2S***

5.2 La visualisation des données d'ARN

5.2.1 Analyse de l'existant

5.2.1.1 Les interfaces graphiques dédiées à l'ARN

Les données produites par les algorithmes, pour les différents niveaux d'étude de l'ARN sont souvent sous forme textuelle. Si l'on veut pleinement les comprendre, il faut disposer, pour chaque niveau, d'interfaces graphiques permettant de les visualiser. Ces interfaces devront proposer un affichage clair des informations biologiques et permettre de les manipuler de façon intuitive. La visualisation et la manipulation des données permettent à l'utilisateur de valider, rejeter ou éditer les résultats des algorithmes, et est donc essentielle. A l'heure actuelle, la majorité des interfaces graphiques dédiées à l'ARN se focalisent sur les alignements et les structures secondaires.

5.2.1.2 La visualisation et l'édition d'alignements

Une grande partie des outils graphiques d'étude de l'ARN sont destinés à l'affichage et la manipulation des alignements. Ces outils peuvent être simples, comme **RALEE** [Griffiths-Jones05a], qui est un module du logiciel d'édition de texte **Emacs**. Il existe également des outils plus complexes comme **Ribostral** [Mokdad06] qui permet de visualiser des alignements dans **MATLAB**. Dans cet outils les séquences sont alignées contre une molécule de référence dont la structure tridimensionnelle est connue. Un score est attribué à chaque paire de bases alignée avec une paire de bases de la structure de référence en fonction du respect de l'isostérie. Ces scores permettent de mettre en valeur la conservation structurale de certains domaines et donc de construire un alignement structural. L'environnement **jPHYDIT** [Jeon05] permet de visualiser et éditer des alignements de séquences d'ARN et de les exploiter pour l'étude de la phylogénie. Les outils **4SALE** [Seibel06] et **SARSE** [Andersen07] permettent d'aligner

des structures secondaires de molécules d'ARN en plus de leurs séquences. **SARSE** possède la fonctionnalité intéressante de pouvoir afficher côte à côte deux vues de l'alignement pour présenter des régions distantes de la séquence.

5.2.1.3 La visualisation de structures secondaires

Il existe un très grand nombre d'algorithmes dont l'objectif est la prédiction de structure secondaire. Un certain nombre d'outils permettant leur visualisation est donc disponible. **RNAstructure** [Mathews07] est un programme de prédiction de structure secondaire capable d'afficher ses résultats de façon graphique. **RnaViz** [De Rijk03] est un outil dont le but est la production de dessin de structure secondaire de qualité, afin de servir d'illustration dans des publications scientifiques. Le logiciel **JViz.Rna** [Wiese05] permet de représenter des structures secondaires de diverses manières (représentation classique, linéaire, circulaire, *Dot Plot*, ...). Certains programmes comme **Pseudoviewer2** [Han03] sont spécialisés dans la représentation de structures secondaires contenant des pseudonœuds.

5.2.1.4 Les environnements visuels intégrés

Afin de permettre la confrontation de diverses données relatives à une molécule d'ARN et ainsi de favoriser leur compréhension, la possibilité d'afficher plusieurs niveaux d'analyses, soit dans une même interface graphique, soit en interconnectant plusieurs interfaces est primordiale.
De plus, la possibilité de contrôler des algorithmes directement par l'interface graphique dédiée à afficher leurs résultats permet à l'utilisateur de ne pas avoir à lancer et gérer de nombreux programmes et les fichiers qu'ils créent.
Le développement de telles interfaces est un travail long et délicat, ce qui explique le faible nombre d'interfaces graphiques de qualité dédiées à l'étude des ARN face à la grande quantité de données à étudier. **ARB** [Ludwig04] est un environnement

bioinformatique d'étude de l'ARN. Les objectifs principaux de **ARB** sont la maintenance d'une base de données intégrative combinant des séquences d'ARN et des données additionelles qui leur sont liées, et la mise à disposition d'une sélection d'outils d'études de l'ARN interconnectés avec cette base. Parmi les analyses permises par les outils disponibles on retrouve notamment les alignements de séquences, l'édition de structures secondaires, et la visualisation et l'édition d'arbres phylogénétiques. Avec une approche analogue, **StructureLab** [Shapiro96] est une infrastructure informatique développée pour permettre l'utilisation d'un large éventail d'approches d'analyses structurales d'ARN. Ce système propose un certain nombre d'outils intégrés et permettant l'utilisation de données de biologies humides dans le but de déterminer la structure d'une molécule donnée d'ARN. **StructureLab** permet de manipuler des séquences et leur alignement, des algorithmes de prédictions de structures, la représentation de structures secondaires et tertiaires, des données taxonomiques, ...

5.2.2 Les interfaces graphiques de P.A.R.A.DIS.E

Au sein de notre équipe, plusieurs interfaces graphiques dédiées à l'étude des ARN ont été développées. Deux de ces interfaces, le visualiseur de structure secondaire **RNA2DViewer** et l'aligneur **RNAlign** ont été interconnectées au sein de l'application **S2S**. Cette application sert de fondation à la plateforme d'analyse des annotations d'ARN **P.A.R.A.DIS.E** qui, en plus de contenir ces outils, donne accès à une nouvelle interface, nommée **Assemble** constituant l'essentiel de mon travail de thèse, et permettant la modélisation de structure tertiaire d'ARN.

5.2.2.1 RNA2DViewer : Un éditeur de structures secondaires

RNA2DViewer est une interface graphique dont le développement a démarré avec le logiciel **RnaMLView** [Yang03], pour être amélioré dans **S2S** [Jossinet05] puis dans **P.A.R.A.DIS.E**. Cet outil a été développé à l'origine pour permettre la visualisation et la manipulation de structures secondaires étendues obtenues par **RNAView** [Yang03] à partir de structures tridimensionnelles d'ARN (voir **5.4.8.1**).


Une ***structure secondaire étendue*** est une structure secondaire dans laquelle figure également les paires de bases non canoniques, définies par la classification *Leontis-Westhof* [Leontis01]. Cette représentation offre donc plus d'informations qu'une structure secondaire classique, sans entrer dans des détails de résolution atomique.

L'algorithme **RNAView** qui fournit la structure secondaire propose également une représentation de cette structure par projection orthogonale de la structure tertiaire. Ce type de projection entraîne un fort recouvrement entre les éléments structuraux, rendant le dessin illisible pour des molécules de grande taille. **RnaMLView** a donc été développé pour permettre à l'utilisateur de naviguer dans le dessin de la structure en lui donnant la possibilité de déplacer et orienter chaque hélice, et donc d'arranger la représentation de la structure secondaire selon un schéma qui lui est familier (par exemple, une feuille de trèfle pour un ARN de transfert).
RNA2DViewer représente les paires de bases à l'aide des symboles de la nomenclature *Leontis-Westhof* (figure **5.2**) et offre la possibilité de masquer certaines paires en fonction de leurs types.
Grâce à l'interconnexion des outils graphiques, au sein de **S2S**, **RNA2DViewer** sert de carte à l'utilisateur pour naviguer dans des structures d'ARN de grande taille, comme les ribosomes. L'utilisateur peut ainsi parcourir la structure secondaire d'une molécule et centrer la vue ou zoomer sur un élément structural particulier, ce qui se répercute par une opération analogue dans les autres vues (visualiseur 3D et aligneur) de **S2S** (figure

5.5(a)). Afin de se repérer plus aisément dans la structure, il est possible de n'afficher que les n plus proches voisins d'un élément structural (par exemple une hélice).
En plus de la position et de l'orientation des hélices dans le dessin, de nouvelles améliorations de cet outil dans **`P.A.R.A.DIS.E`** permettent à l'utilisateur d'éditer directement la structure secondaire, en supprimant ou ajoutant des hélices ou des paires de bases de tout types. De plus, de nombreux algorithmes détaillés plus loin, permettent d'obtenir une structure secondaire à partir d'une séquence (voir **5.4.8.3**), un dessin de structure secondaire non recouvrant (voir **5.4.8.4**) pour une structure secondaire, ou encore un premier jet de structure tertiaire à partir d'une structure secondaire (voir **7.2.3**).

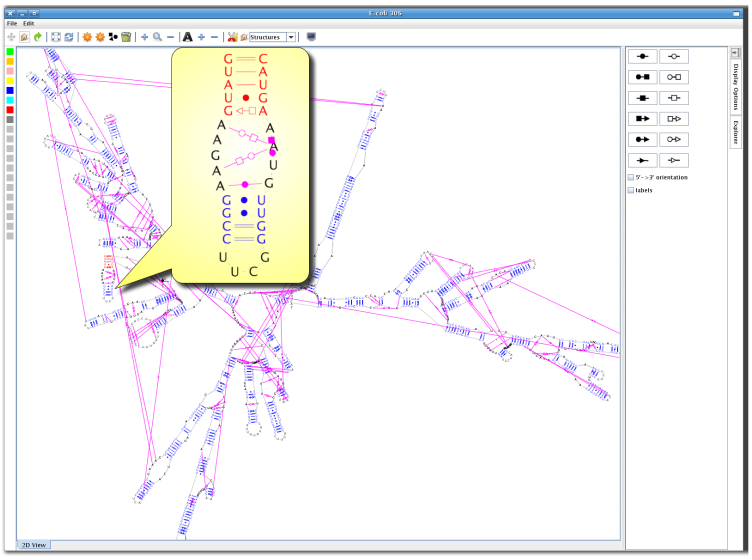


FIG. 5.2 – **`RNA2DViewer`**
La structure secondaire de la grande sous unité ribosomale de E.coli dessinée de façon non recouvrante dans ***`RNA2DViewer`****. Les paires de bases sont représentées à l'aide de la nomenclature Leontis-Westhof.*

5.2.2.2 RNAlign : Un aligneur de sequences

RNAlign a été développé pour étudier les familles d'ARN orthologues sur la base d'alignements structuraux et pour permettre à l'utilisateur d'éditer et construire ces alignements. Cet outil, déjà présent dans le logiciel **S2S**, indique la conservation de séquence de manière classique en attribuant une couleur à chaque type de nucléotide. L'avantage majeur de **RNAlign** sur la plupart des autres aligneurs est l'incorporation d'informations structurales. En effet, l'une des molécules de l'alignement, dont la structure secondaire étendue est connue, est utilisée comme molécule de référence. Cette structure de référence est affichée en notation parenthésée.


La ***notation parenthésée*** permet une représentation textuelle d'une structure secondaire en symbolisant les nucléotides non appariés par des points, les paires de bases canoniques par des couples de parenthèses ouvrantes et fermantes et les paires de bases non canoniques par des couples de symboles *inférieur* et *supérieur*.

RNAlign affiche un alignement structural en indiquant la compatibilité de chaque séquence avec la structure de référence. Pour évaluer cette compatibilité, **RNAlign** incorpore les matrices d'isostérie des paires de bases de la classification *Leontis-Westhof* [Leontis02b]. Lorsqu'une paire de bases de la structure de référence est alignée avec deux nucléotides d'une séquence, ces nucléotides sont affichés dans une certaine couleur pour indiquer si une paire de bases du même type peut être formée avec ces nucléotides et si la paire formée est isostérique avec celle de la structure de référence. Par ailleurs l'interconnexion des outils de **S2S** permet d'analyser et d'éditer l'alignement en gardant sous les yeux la structure secondaire et tertiaire (si elle est disponible) de la molécule de référence.

Dans **P.A.R.A.DIS.E**, de nombreuses améliorations ont été apportées à cet outil. La première est la possibilité d'afficher dans la même fenêtre plusieurs vues du même alignement. Cette fonctionalité, qui n'existe pas dans la plupart des aligneurs,

est particulièrement utile pour observer simultanément des régions distantes d'un alignement entre des molécules de grande taille et l'effet des éditions sur ces régions. Certaines de ces vues pourront afficher les séquences dans le sens 3' vers 5'. Les paires de bases étant affichées entre deux vues successives à l'aide des symboles de la nomenclature *Leontis-Westhof*, cette symétrie entre les vues permet de mettre en valeur les hélices (figure **5.3**).

Si la structure secondaire de référence a été déduite d'une structure tridimensionnelle, cette dernière peut servir à générer la structure des autres molécules par homologie. En effet, lorsque deux régions sont alignées et présentent une grande conservation structurale, les repliements tridimensionnels de ces régions sont vraisemblablement très proches. Il est donc possible, pour une région donnée de copier le repliement de la molécule de référence et de l'appliquer à la région homologue d'une autre molécule. L'application de la conformation se fait par simple substitution de bases entre les nucléotides de la molécule de référence et la séquence de la molécule générée.

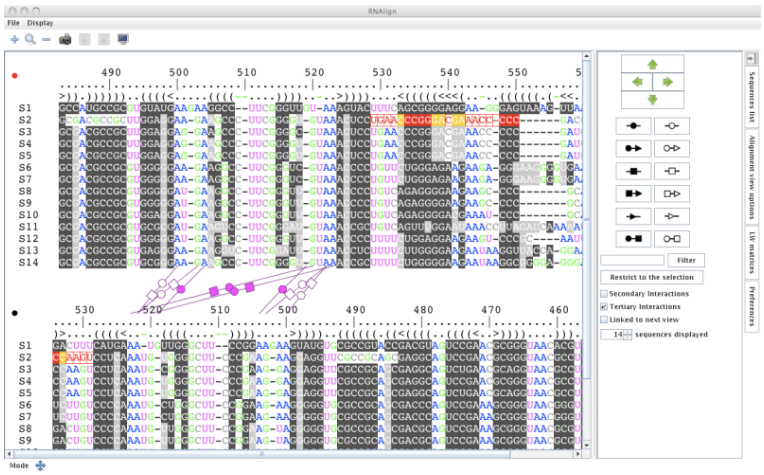


FIG. 5.3 – **`RNAlign`**

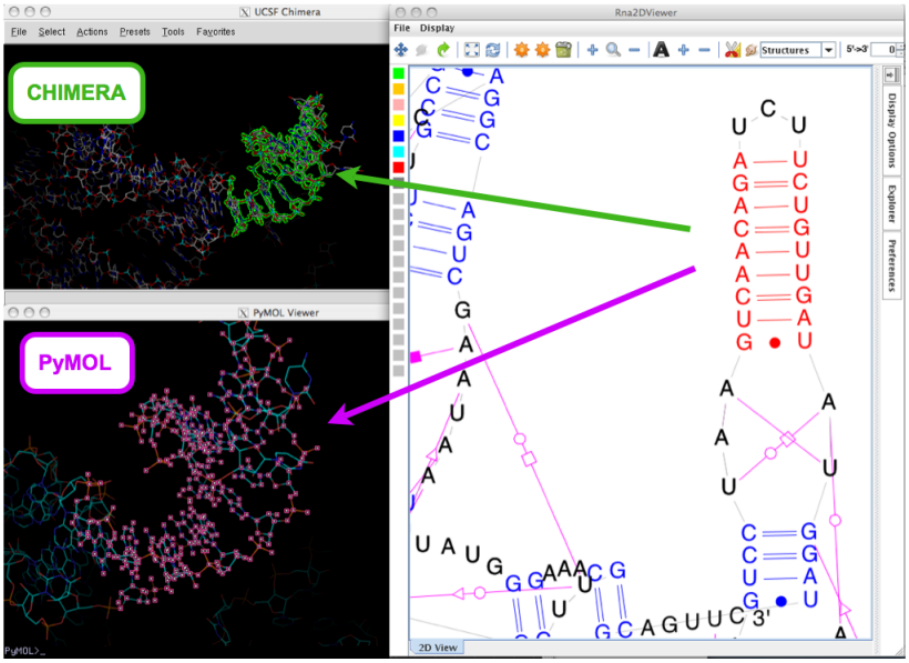
Un alignement structural représenté dans ***`RNAlign`*** *à l'aide de deux vues, l'une dans le sens 5' vers 3', l'autre dans le sens opposé. Les symboles de la nomenclature Leontis-Westhof sont utilisés entre ces deux vues pour symboliser les paires de bases canoniques ou non. Les paires de bases isostériques à celles de la structure de référence sont représentées par des carrés noirs. Des carrés gris indiquent des paires de bases géométriquement possibles mais non isostériques.*

5.2.2.3 External Tertiary Viewer : Un visualiseur de structures tertiaires

Dans **S2S**, l'utilisateur peut visualiser les structures tridimensionnelles des molécules qu'il étudie grâce au visualiseur 3D externe **PyMOL** [pymol]. Ce visualiseur est interconnecté aux autres outils de **S2S** et les actions de l'utilisateur dans l'un d'eux (**Rna2DViewer** ou **RNAlign**) se répercutent dans **PyMOL**.
L'outil nommé **External Tertiary Viewer** (figure **5.4**) permet de contrôler cette interconnexion. L'action à effectuer dans le visualiseur est personnalisable, et l'utilisateur peut choisir de centrer ou zoomer sur les nucléotides sélectionnés ou bien d'afficher ou cacher uniquement la sélection. Cet outil agit donc comme un traducteur entre les messages produits par les outils de **P.A.R.A.DIS.E** et les visualiseurs externes qui possèdent chacun leur propre langage de communication.
Dans **P.A.R.A.DIS.E**, cette interconnexion a été étendue aux visualiseurs 3D **Chimera** [Huang96] et **VMD** [Humphrey96].
Cet outil nous permet donc d'exploiter des fonctionalités de ces visualiseurs externes qui sont actuellement absentes de nos outils. Par exemple, il est possible d'afficher des ions et des solvants, d'utiliser divers modes de rendu pour les molécules et les cartes de densités, de faire des films de qualité grâce au lancé de rayons (*Raytracing*), ...

5.2.2.4 Assemble : Un modélisateur moléculaire

Comme nous venons de le voir, **S2S** permet d'annoter une structure tertiaire par une structure secondaire étendue, et d'utiliser cette structure secondaire comme structure de référence pour l'alignement structural de séquences orthologues d'ARN. Une fois cet alignement construit, il faudrait pouvoir construire une structure tertiaire pour une ou plusieurs molécules orthologues. **Assemble** est l'outil de **P.A.R.A.DIS.E** dédié à la modélisation moléculaire et propose de nombreux automatismes et fonctionalités dont le but et de rendre cette tâche la plus rapide et aisée possible. Le développement d'**Assemble** et son intégration à l'infrastructure **P.A.R.A.DIS.E** représentant



(a) Les visualiseurs externes

(b) External Tertiary Viewer

FIG. 5.4 – La visualisation de structures tertiaires dans **P.A.R.A.DIS.E**
(a) Les visualiseurs externes (ici ***PyMOL*** *et* ***Chimera****) sont interconnectés avec* ***P.A.R.A.DIS.E*** *et réagissent aux évènements des autres interfaces graphiques. (b) L'****External Tertiary Viewer*** *permet de contrôler cette interconnexion en précisant la nature du message envoyé (voir/cacher, centrer, zoomer,...) et le visualiseur devant afficher une structure tertiaire donnée.*

l'essentiel de mon travail de thèse, le chapitre **7** est entièrement dédié à cet outil et aux algorithmes qu'il utilise.

5.2.2.5 L'interconnexion des outils graphiques de **P.A.R.A.DIS.E**

Les différentes interfaces graphiques de **P.A.R.A.DIS.E** sont interconnectées et toute action de l'utilisateur (sélection, zoom, ...) dans l'une d'entre elles prévient les autres outils qui peuvent réagir en fonction de leur état. Cette interaction repose sur un échange de messages entre les interfaces graphiques. Les messages échangés ont une sémantique très simple, car il ne s'agit que d'une liste de nucléotides sélectionnés (l'action sur ces nucléotides est déterminée par l'état de l'outil destinataire du message). De plus ils sont unidirectionnels (c'est à dire que l'émetteur du message n'attend pas de réponse de la part des destinataires). La mise en place d'un protocole complexe de communication n'est donc pas nécessaire.
Comme ce fut le cas dans **S2S**, la communication entre les outils de **P.A.R.A.DIS.E** repose sur un système événementiel inspiré du patron de conception (*design pattern*) observateur (*Observer*). Lorsqu'un nouvel outil graphique est lancé, celui-ci est ajouté à la liste des outils à prévenir en cas d'actions de l'utilisateur. Lorsqu'une action est réalisée dans un outil, un message est envoyé à tous les autres, qui réagissent suivant l'état dans lequel ils se trouvent (zoom, simple sélection, ...).
Les messages entre les interfaces graphiques sont assez simples, car ils ne font référence qu'à des nucléotides et qu'indépendamment du niveau d'analyse manipulé par l'outil graphique, la notion de «nucléotide» a toujours un sens. Cependant, lorsque que l'on cherche à autoriser une communication, sémantiquement riche, entre outils hétérogènes (interfaces graphiques et algorithmes manipulant différents niveaux d'analyse), le développement d'un langage plus complet, représentant l'ensemble des concepts biologiques manipulés et leurs relations, est nécessaire.

5.3 La modélisation informatique des concepts biologiques liés à l'ARN

La mise en place d'une infrastructure logicielle permettant l'analyse de données biologiques nécessite au préalable le développement d'un système informatique visant à modéliser de la façon la plus complète et fidèle possible les concepts biologiques étudiés et facilitant la communication entre les différents modules de cette infrastructure. De nombreux groupes ont travaillé sur la mise en place de tels modèles.

5.3.1 Analyse de l'existant

La **O|B|F** (Open Bioinformatics Foundation) est une organisation visant à encourager le développement de projets bioinformatiques *open source*. Cette initiative est née de trois projets indépendants : **BioPerl** [bioperl], **BioJava** [biojava] et **BioPython** [biopython] et forment avec **BioRuby** [bioruby], **BioDas** [biodas] et bien d'autres les projets **Bio*** [Mangalam02]. Les projets **Bio*** ont pour but de fournir, pour un grand nombre de langages de programmation et de technologies, des librairies pouvant servir de base à des projets bioinformatiques.
La librairie de programmation **BioJava** et son extension **BioJavaX** fournissent un modèle informatique de concepts biologiques, exploitant rigoureusement les avantages de la programmation orientée objet. Il est possible grâce à ce modèle de décrire des séquences et les propriétés biologiques qui leur sont associées, à l'aide d'un système d'annotation de séquences. Cette librairie permet également de modéliser des alignements, de mettre en place des algorithmes reposant sur la programmation dynamique et de développer des interfaces graphiques dédiées à l'analyse de séquence. **BioJava** a été utilisé dans plusieurs projets bioinformatiques *open source* [biojavainside], notamment **Bioclipse** [Spjuth07] une plateforme graphique

d'analyse bioinformatique, **Cytoscape** [Shannon03] qui permet la visualisation de réseaux d'interactions moléculaires et **BioWeka** [Gewehr07] qui permet de faire de la fouille de données biologiques.
Le projet **BioPerl** [Stajich02] est une collection de modules **Perl** dédiés à la bioinformatique qui connaît un important succès et a été utilisé dans de nombreux projets de grande envergure comme la banque de données **EnsEMBL** [Flicek08].
De façon analogue à **Bio***, **BTL** (Bioinformatics Template Library) [Pitt01] vise à faciliter et à accélérer le développement de programmes bioinformatiques efficaces en **C++**. Pour ce faire, **BTL** fournit une librairie de concepts et d'algorithmes fréquemment utilisés en bioinformatique, sous une forme générique qui permet de les combiner de façon flexible dans des programmes utilisant la programmation orientée objet. Le **NCBI C++ Toolkit** [ncbitoolkit] propose également un environnement de programmation **C++** pour construire des modèles biologiques, à l'aide d'une librairie de concepts, de contacter des algorithmes tels que **BLAST** ou **ORF finder** et de développer son propre environnement logiciel (incluant des fonctionnalités graphiques, réseaux, bases de données, ...).
Le but du NCICB (National Cancer Institute Center for Bioinformatics) est de fournir des infrastructures informatiques utiles à la recherche sur le cancer. Le projet **caBIO** [Covitz03] a initialement été développé pour permettre l'accès aux informations de divers projets du NCICB. Son succès a conduit à son adoption comme architecture principale pour l'unification et l'intégration des données du NCICB. **caBIO** se divise en trois couches :

1. la couche de données, qui permet de faire appel aux diverses sources d'informations (fichiers, bases de données, ...)
2. la couche objet, qui modélise les multiples concepts utilisés en recherche sur le cancer. Cette couche est développée en faisant appel à la programmation orientée objet et peut être étendue pour intégrer des concepts appartenant à l'ensemble des domaines de la biologie.

3. la couche de présentation, qui permet de distribuer les informations *via* des serveurs web

La librairie **caBIO** est notamment utilisée dans l'infrastructure **caCORE** décrite en section **5.4**

5.3.2 Le modèle de concepts biologiques de P.A.R.A.DIS.E

Le modèle objet de l'infrastructure **P.A.R.A.DIS.E** peut être vu comme un «*moteur d'annotation des molécules d'ARN*» et a pour but de servir de couche intermédiaire entre les objets manipulés dans les modules graphiques et les données brutes, produites par les algorithmes. Comme le modèle de **BioJava** dont il s'inspire, il repose sur le paradigme qu'une molécule d'ARN peut être annotée en un ou plusieurs nucléotides, contigus ou non, par certaines informations. Ceci peut être traduit par la relation suivante entre les concepts fondamentaux de notre modèle :

 Une *Annotation* relie un *Feature* à une *Location* d'une *Molecule*

Dans notre moteur d'annotation, les différents concepts liées à l'ARN sont donc des *Feature*. Bien qu'inspiré du modèle de **BioJava**, celui de **P.A.R.A.DIS.E** présente une différence essentielle, en autorisant une même information biologique à annoter plusieurs molécules. Ainsi, une paire de bases intermoléculaire ou un alignement de séquences sont des propriétés qui affectent plusieurs molécules. Pour refléter cette réalité biologique, un même *Feature* peut être associé à plusieurs *Molecule*. Chaque couple *Feature*/*Molecule* est relié par une objet de type *Annotation* qui précise également la *Location* de la *Molecule* annotée par le *Feature*.

Grâce à la propriété d'«héritage» de la *POO*, un *Feature* peut se spécialiser. Par exemple, une *paire de bases cis Hoogsteen-Hoogsteen* est un type particulier de *paire*

de bases.

De plus, une relation parent-enfant (permise par la propriété de «composition» de la POO) pourra être mise en place entre certains `Feature` afin de refléter :

1. qu'un `Feature` est composé d'un autre `Feature` (par exemple, une structure secondaire est composée d'hélices)
2. qu'un `Feature` est issu de l'analyse d'un autre `Feature` (par exemple, une structure secondaire pourra être déduite d'une structure tertiaire par un algorithme d'annotation)

Le tableau **5.2** présente la liste actuelle de ces concepts, leurs possibles liens de parenté, le concept qu'ils spécialisent et leur types de `Location` compatibles. Les figures **5.5** et **5.6** illustrent les différentes propriétés du moteur d'annotation de **P.A.R.A.DIS.E** en représentant des molécules, leurs propriétés structurales et leur modélisation au sein de notre infrastructure.

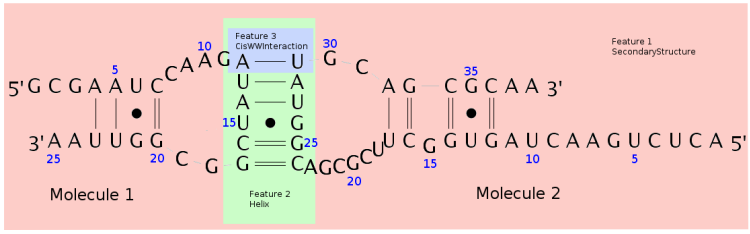


FIG. 5.5 – Les propriétés structurales de deux molécules

La structure secondaire de deux molécules. Ces molécules forment une hélice intermoléculaire (en vert). Cette hélice est composée, entre autre, d'une paire de base cis Watson-Crick/Watson-Crick (en bleu).

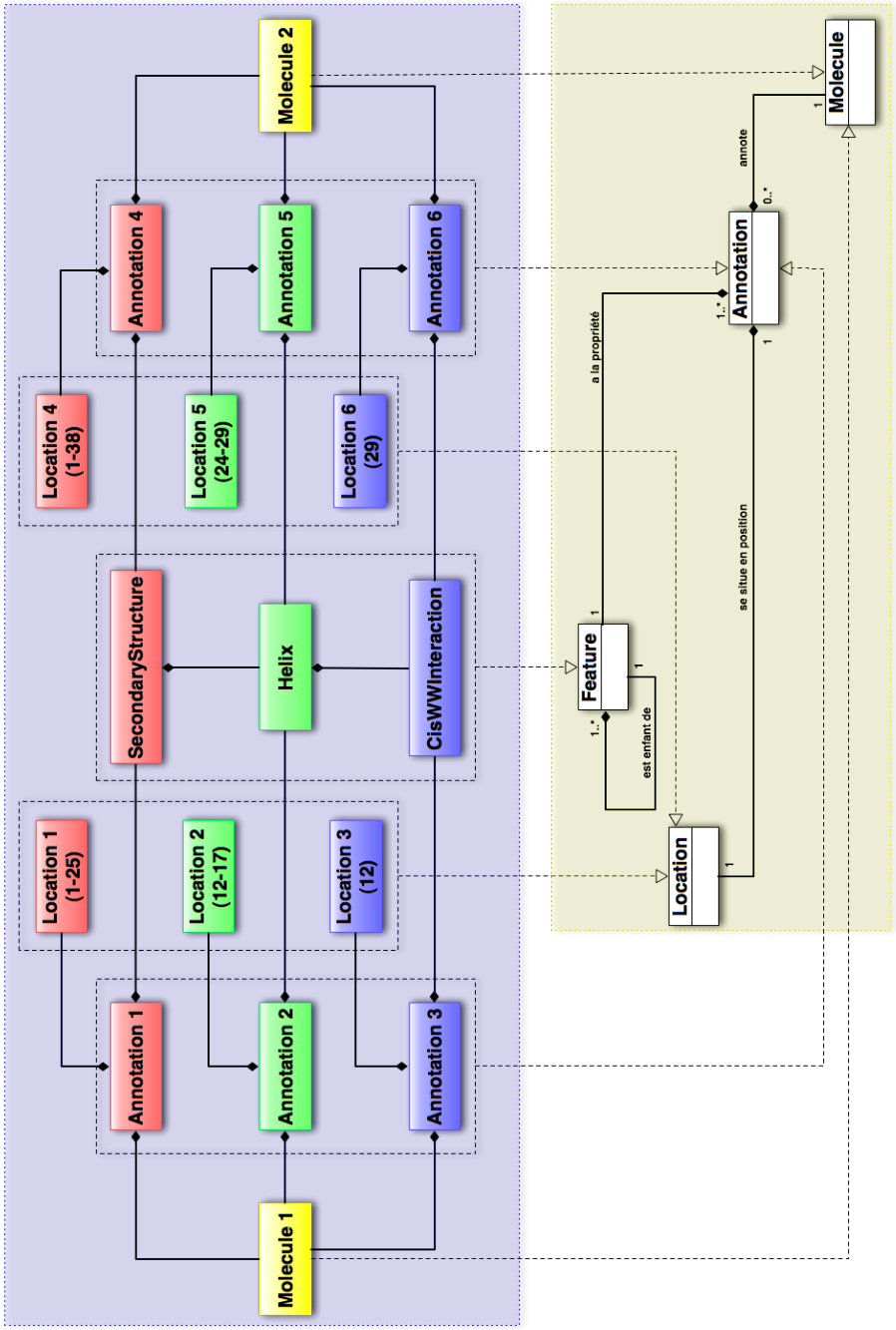


FIG. 5.6 – Les concepts de **P.A.R.A.DIS.E**

*Dans l'encart jaune : le diagramme UML des concepts de **P.A.R.A.DIS.E***
Dans l'encart bleu : les instances représentant les données de la figure **5.5**

Feature	Description	Parents possibles	Spécialise	Location
TertiaryStructure	structure tertiaire	*SecondaryStructure*		1 intervalle sur $n \geq 1$ molécules
*Residue3D**	jeu de coordonnées 3D des atomes d'un résidu	*TertiaryStructure*		1 résidu
SecondaryStructure	structure secondaire	*TertiaryStructure*		1 intervalle sur $n \geq 1$ molécules
StructuralDomain	domaine structural	*SecondaryStructure*		n intervalles sur m molécules
SingleStrand	simple brin	*SecondaryStructure*	*StructuralDomain*	1 intervalle
Helix	double hélice d'ARN	*SecondaryStructure*	*StructuralDomain*	2 intervalles sur 1 ou 2 molécules
PseudoKnot	pseudonœud	*SecondaryStructure*	*StructuralDomain*	1 intervalle
StructuralInteraction	interaction entre deux résidus	*SecondaryStructure*		2 résidus sur 1 ou 2 molécules
*BaseBaseInteraction**	interaction entre deux bases	*Helix*, *SecondaryStructure*	*StructuralInteraction*	2 résidus sur 1 ou 2 molécules
AtomAtomInteraction	interaction entre deux atomes	*BaseBaseInteraction*, *SecondaryStructure*	*StructuralInteraction*	2 résidus sur 1 ou 2 molécules

Feature	**Description**	**Parents possibles**	**Spécialise**	**Location**
`SecondaryStructureDisplay`	dessin de structure secondaire	`SecondaryStructure`		1 intervalle sur $n \geq 1$ molécules
`Residue2D`	coordonnées 2D d'un résidu dans un dessin	`SecondaryStructureDisplay`		1 résidu
`StructuralAlignment`	alignement structural	`SecondaryStructure`		1 intervalle sur $n \geq 2$ molécules
`Identity`	identité dans un alignement	`StructuralAlignment`		voir les spécialisations
`ResidueIdentity`	identité entre plusieurs résidus	`StructuralAlignment`	`Identity`	1 résidu de $n \geq 2$ molécules
`StructuralIdentity`	identité entre plusieurs domaines structuraux	`StructuralAlignment`	`Identity`	1 intervalle sur $n \geq 2$ molécules
`TertiaryMotif`	motif de structure tertiaire	`SecondaryStructure`, `TertiaryStructure`		1 intervalle sur $n \geq 1$ molécules

TAB. 5.2 – La liste des `Feature` de **P.A.R.A.DIS.E**

Les `Feature` ***marqués par * sont spécialisés (il existe une spécialisation de*** `Residue3D` ***par type de résidu, et une spécialisation de*** `BaseBaseInteraction` ***pour chaque type d'interaction dans la nomenclature Leontis-Westhof). Dans la colonne* Location *un intervalle est une série de nucléotides successifs.***

Il est à noter que certains automatismes ont été introduits dans ce moteur d'annotation des ARN. En effet, lorsqu'un `Feature` est ajouté comme enfant d'un autre, la `Location` du parent est adaptée pour englober celle de l'enfant (par exemple, l'ajout d'une paire de bases en bout d'une hélice étend la `Location` de l'hélice). De plus, la présence de certaines annotations implique la présence d'autres annotations, qui sont automatiquement ajoutées par le système. Par exemple, lorsqu'un `Feature` de type `Helix` est ajouté afin de décrire la présence d'une hélice, des `Feature` de type `cisWWInteraction` symbolisant des paires de bases canoniques sont ajoutés automatiquement et rattachés à l'hélice. De même, l'ajout d'une paire de bases canonique entre une Guanine et une Cytosine implique l'ajout de `Feature` de type `AtomAtomInteraction` pour symboliser les liaisons hydrogènes entre les atomes O6, N1 et N2 de la Guanine et les atomes N4, N3 et O2 de la Cytosine. L'ajout d'hélices qui se «croisent» entraîne l'ajout de `PseudoKnot` pour représenter le pseudonœud.
Le modèle informatique sur lequel repose **P.A.R.A.DIS.E** permet donc d'unifier les concepts utilisés par les interfaces graphiques et algorithmes et leur sert de langage commun. Cependant, il ne peut y avoir compréhension que si ce langage repose sur des moyens de communication permettant d'échanger des messages. Alors que la communication entre les interfaces graphiques repose sur un simple système événementiel, celle impliquant les algorithmes doit répondre à des exigences plus drastiques. Les algorithmes étant souvent difficiles à installer et maintenir et pouvant se montrer exigeants en ressources, il est souhaitable de découpler la partie graphique de la partie algorithmique afin de pouvoir installer les algorithmes sur un serveur puissant et les interfaces graphiques sur l'ordinateur personnel des utilisateurs. Le protocole de communication doit donc être capable de fonctionner sur un réseau.

5.4 La distribution d'infrastructures logicielles sur un réseau

5.4.1 Analyse de l'existant

Il existe de nombreuses technologies permettant de mettre en place une architecture distribuant des ressources algorithmiques. Parmi elles, on retrouve notamment les systèmes de grilles informatiques (*Grid computing*), les infrastructures *pair à pair*, les *services web* et les systèmes multi-agent. Quelques projets illustrant ces techniques sont détaillés dans cette section.

5.4.1.1 Les grilles informatiques

Les grilles informatiques (*Grid Computing*) ont pour but de décomposer un calcul coûteux en des multiples tâches plus petites qui seront exécutées en parallèle sur un nombre plus ou moins important d'ordinateurs. Les résultats de ces opérations seront ensuite intégrés afin de produire le résultat global du calcul initial. Cette approche est utilisée en bioinformatique. **`ABCGrid`** [Sun07] permet d'utiliser les programmes **`NCBI BLAST`**[Johnson08], **`Hmmpfam`**[Eddy98] et **`CE`**[Shindyalov98] sur une grille d'ordinateurs utilisant Windows, Linux ou Mac OS X. **`Grid-Allegro`** [Andrade07] est une version d'**`Allegro`**[Gudbjartsson00, Gudbjartsson05] exécutable sur une grille. Toutefois, les grilles ne sont pas adaptées à la distribution des algorithmes que nous souhaitons appliquer à **`P.A.R.A.DIS.E`** et ne seront pas détaillées plus avant.

5.4.1.2 Les systèmes pair à pair

La technologie pair à pair (*peer-to-peer*, P2P) exploite la multiple connectivité entre les participants d'un réseau pour optimiser le partage de ressources (données, applications, ressources physiques, ...) en faisant abstraction d'un serveur centralisé. **Chinook** [Montgomery05] est une infrastructure bioinformatique reposant sur la technologie P2P. Le but de cette plateforme est de faciliter l'échange de techniques d'analyse (applications) sur un réseau. **Chinook** fonctionne en transformant en services des outils en ligne de commande, et en les diffusant sur un réseau virtuel. Les avantages de cette approches sont que la maintenance des outils sont à la charge de leurs fournisseurs et non des utilisateurs. La redondance de certains programmes au sein d'un même réseau permet de paralléliser les analyses de grandes quantités de données. **Chinook** peut également être utilisé pour comparer la qualité de plusieurs outils équivalents.

5.4.1.3 Les services web

Parmi les technologies d'architectures distribuées les plus fréquemment utilisées en bioinformatique on retrouve les «*services web*». Un service web est un système informatique permettant l'interopérabilité des ressources logicielles et matérielles sur un réseau. Ce système permet la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. La communication repose sur l'échange de fichiers XML. Suivant la grammaire et les protocoles utilisés il existe plusieurs types de services web, les plus utilisés, notamment en bioinformatique, étant les services SOAP (Simple Object Access Protocol) :
L'EBI (European bioinformatics institute) propose de nombreux services web [Labarga07] notamment au travers de deux initiatives :

- **SOAPlab** [Senger03, soaplab] qui permet d'accéder à la majeure partie des outils de **EMBOSS** [Rice00] *via* des services web.

– **openQBS** [openqbs] permettant l'accès à la banque de références bibliographiques **MEDLINE**.

La NCBI (National Center for Biotechnology Information) propose également d'accéder à ses services **Entrez Utilities** à l'aide du protocole SOAP [esoap].

La DDBJ (DNA Data Bank of Japan) a développé le projet **XML Central** [xmlcentral] qui propose des services web permettant d'utiliser des algorithmes tels que **BLAST**, **ClustalW**, **Ensembl**, **SRS**,...

Les services web du KEGG (Kyoto Encyclopedia of Genes and Genomes), **KEGG API** [keggapi], donnent accès à de nombreux outils et bases de données (ayant trait aux voies métaboliques et régulatrices ainsi qu'à des données génomiques et chimiques).

PathPort [Eckart03, pathport] du VBI (Virginia Bioinformatics Institute) donne accès, entre autres, aux outils **Mummer**, **Fasta**, **Glimmer** ainsi qu'à des informations sur les pathogènes et des algorithmes permettant de traiter les données de puces à ADN.

Pour des revues récentes sur les principaux services web dans le domaine des sciences de la vie, voir [Curcin05, Neerincx05, Foster05].

Les services web permettent donc d'accéder programmatiquement à des ressources disponibles sur internet. Des environnements logiciels ont vu le jour afin de pouvoir visualiser et manipuler le flot de données associées à ces services web. Parmi ces projets on notera particulièrement **caCORE**, **BioMOBY**, ^{my}**Grid** et **Taverna**

caCORE [Covitz03, Komatsoulis08] est un projet du NCICB (National Cancer Institue of BioInformatics) visant à intégrer les services bioinformatiques utiles pour la recherche sur le cancer et la médecine. Toutefois, l'architecture **caBIO** au cœur de **caCORE** peut être étendue à n'importe quel domaine de la bioinformatique (voir la section **5.3**). Les services reposant sur **caBIO** peuvent être contactés *via* SOAP et le système peut lui-même utiliser SOAP pour contacter des services externes.

BioMOBY [Wilkinson02] est un projet open source visant à fournir un système de découverte et de traitement de données biologiques en permettant à un laboratoire d'enregistrer les services web qu'ils proposent au sein d'un annuaire appelé MOBY central. La philosophie de **BioMOBY** est de proposer à chaque étape les différentes

analyses disponibles en fonction des données dont l'utilisateur dispose. Il est également possible de proposer à l'utilisateur une succession complexe d'analyses lui permettant d'obtenir un certain type de données en fonction des données qu'il possède actuellement. **BioMOBY** est désormais en version 1.0 et donne accès à plus de 1400 ressources bioinformatiques [Wilkinson08].
^{my}**Grid** [Stevens03] est un projet développé pour accéder à plusieurs types de services en utilisant **Java** et **SOAP**. Contrairement à **BioMOBY**, ^{my}**Grid** n'est pas en mesure de découvrir automatiquement des enchaînements d'analyses, mais peut être dirigé par une infrastructure logicielle appelée **Taverna**.
Taverna [Oinn04, Lanzén08] permet de créer des enchaînements d'analyses pour ^{my}**Grid** et permet d'intégrer plusieurs types de services dont ceux de **SOAPlab** et de **BioMOBY** [Kawas06]. Pour décrire ces enchaînements **Taverna** utilise son propre langage : le Scufl (Simple Conceptual Unified Flow Language).
Enfin, des outils tels que **Cyrille2** [Fiers08], **GPIPE** [Garcia Castro05], **Pegasys** [Shah04], **Wildfire** [Tang05] et **MOWserv** [Navas-Delgado06] permettent, à l'aide d'interfaces graphiques ou web, de mettre en place des enchaînements d'analyses bioinformatiques en utilisant des services web.

Malgré le succès actuel des services web, cette technologie ne semblait pas présenter la meilleure solution pour l'implémentation de notre architecture distribuée.
L'avantage majeur des services web est de pouvoir faire abstraction, de chaque coté, du langage de programmation. Toutefois, **P.A.R.A.DIS.E** étant développé intégralement en **Java**, l'échange d'objets entre différentes machines peut se reposer sur la sérialisation. La sérialisation permet de représenter les objets sous forme binaire, ce qui permet de créer des messages beaucoup plus léger qu'avec une représentation textuelles, et notamment le format XML qui est particulièrement volumineux. **Java** offre naturellement cette possibilité, grâce à la technologie **RMI** (Remote Method Invocation).
En plus de la taille importante des messages XML, entraînant une lenteur dans

la communication, la mise en place d'un service web est relativement fastidieuse. Les différentes technologies permettant leur implémentation sont souvent dirigées techniquement par de grands groupes commerciaux plutôt que par des initiatives open source. SOAP évolue rapidement, et le changement de version de SOAP entraîne souvent une incompatibilité du code écrit pour une version précédente. Il est donc nécessaire de modifier ou réécrire une partie du code à chaque changement de version, rendant la maintenance d'un tel système très lourde. En conséquence, la technologie que nous avons retenue pour développer notre architecture distribuée et celles des systèmes multi-agent.

5.4.2 Les architectures multi-agent : application à l'infrastructure `P.A.R.A.DIS.E`

La modélisation de systèmes complexes nécessite l'utilisation de techniques permettant la représentation la plus fidèle possible des concepts étudiés. De manière analogue à la POO qui est adaptée pour modéliser les concepts biologiques et leurs relations, la technologie multi-agent fournit une solution efficace pour modéliser les interactions, l'organisation et la coopération des producteurs (algorithmes) et consommateurs (interfaces graphiques) de ces données biologiques. Cette approche intuitive, permettant d'exprimer à haut niveau les fonctionalités d'un système, tout en procurant une conception modulaire et un mécanisme flexible d'interaction, rend les systèmes multi-agent adaptés au développement d'architectures distribuées [Karasavvas04].

Un **système multi-agent** (MAS) est un environnement composé de multiples agents «intelligents» interagissants. Ce type de système possède les caractéristiques suivantes

- **Autonomie** : les agents sont, au moins partiellement, autonomes
- **Vues locales** : aucun agent n'a une vue globale du système, ou ce dernier est trop complexe pour être entierment analysable par un unique agent

– **Décentralisation** : il n'existe pas d'agent responsable de la gestion de tous les autres

Chaque agent réagit aux stimuli de son environnement, par un comportement défini par sa programmation, et agît en retour pour modifier cet environnement.
Les agents peuvent partager leur connaissances en communiquant à l'aide de protocoles tels que le *KQML* (Knowledge Query Manipulation Language) ou l'*ACL* (Agent Communication Language).

5.4.2.1 Les MAS en bioinformatique

Les MAS sont de plus en plus utilisés en bioinformatique [Keele05]. Ces projets utilisent principalement les MAS afin de décomposer la résolution d'un problème en sous tâches affectées chacune à un agent. La capacité de communication des agents est alors exploitée afin d'intégrer leurs résultats et de produire une réponse globale au problème. Le but du MAS proposé par [Armano07] est d'aider à identifier des publications susceptibles d'intéresser un utilisateur donné, en parcourant les données de BMC Bioinformatics [bmcbioinformatics] et Pubmed [pubmed]. La décomposition du problème se fait en attribuant une couche pour chacune des tâches suivantes : gérer les sources d'informations de natures différentes, encoder l'information contenue dans le texte pour la classer dans des catégories, décider si la publication est pertinente en fonction des intérêts de l'utilisateur et demander l'avis de l'utilisateur sur la publication. Un agent est affecté à chaque couche, et la communication se fait, entre deux couches successives, à l'aide d'agents intermédiaires. **AMELIE**[Vignal96, Vignal97, Vignal99] est un MAS développé pour détecter les sites d'épissage dans des séquences. La détection des sites (donneurs ou accepteurs) d'épissage repose sur la satisfaction de certains critères, classés du grain le plus gros au plus fin. La décomposition du problème se fait en affectant un agent à la vérification de la satisfaction de chaque critère. Lorsqu'un agent constate qu'un critère est satisfait, il demande à l'agent charger de vérifier le critère suivant (de grain plus fin) de s'exécuter.

5.4.2.2 Les MAS pour permettre l'intégration de données

Les MAS sont utilisés notamment pour mettre en place des systèmes d'intégration [Karasavvas02]. **`BioAgent`** [Angeletii01] est un MAS dédié à l'intégration de données issues de puces à ADN. Ces données sont réparties sur un grand nombre de bases de données utilisant des formats variés. **`BioAgent`** propose d'affecter un agent par base, servant de traducteur entre le format de celle-ci et un format standard, et d'utiliser la capacité de communication des agents pour intégrer ces différentes données. **`GeneWeaver`** [Bryson00a] est un MAS dédié à l'analyse de génomes et à leur annotation par des données protéiques structurales et fonctionnelles. Dans ce système, de nombreuses ressources sont accessibles (bases de données génomiques ou structurales, algorithmes, navigateur génomique) et un agent est responsable de la gestion de chaque ressource. La communication entre les agents permet la production de résultats par l'enchaînement d'analyses. **`BioMas`** [Decker02] est un système multi-agent pour l'annotation automatique, et le stockage dans des banques, de données de séquence. Le système se soustrait aux analyses manuelles et rend les annotations disponibles pour les chercheurs et les autres MAS. Les agents sont répartis en trois types responsables de l'interaction avec les banques de données, l'interaction avec les utilisateurs et le traitement des données. De plus, chaque agent appartient à un ou plusieurs des quatre sous-systèmes multi-agent chargés de l'intégration de bases de données d'annotation de séquences, de gérer les requêtes provenant de l'interface web, d'utiliser la Gene Ontology pour déterminer la fonction d'un gène, et de produire de nouvelles séquences à annoter.

5.4.2.3 Les MAS comme système de simulation

De nombreuse initiatives ont été prise pour utiliser les MAS à des fins de simulations de processus biologiques [Merelli07]. En effet, dans les systèmes biologiques, les individus sont autonomes, mobiles et réagissent en fonction de leur environnement. Ce

comportement est analogue à celui des agents d'un MAS, ce qui semble indiquer que cette technologie est particulièrement adaptée à ce type de simulation.
Une plateforme de modélisation et d'analyse de systèmes biologiques est présentée dans [Ren08]. Celle-ci a été utilisée pour implémenter un réseau d'interactions protéine-protéine. Dans ce système, chaque protéine est modélisée par un agent. Les interactions entre ces molécules, modélisées à partir de données biologiques, forment des petits réseaux d'interactions. Au cours de la simulation, ces petits réseaux accomplissant une tâche simple commencent à interagir et forment des réseaux plus grands, accomplissant des tâches plus complexes. L'étude de l'évolution entre plusieurs réseaux d'interactions permet d'étudier des phénomènes tels que la duplication et la délétion de gènes.
Des MAS permettent également la simulation de la transduction de signaux [Khan03]. Dans ce système, chaque «espèce» moléculaire est représentée par un agent. Les réactions entre ces molécules se font par l'échange de messages, provoquant la formation du produit de cette réaction (sous forme d'un nouvel agent) et la disparition des agents représentant les réactants. Ce système simule le déplacement des molécules au sein de leurs compartiments cellulaires selon un mouvement brownien et une interaction se forment lorsque deux molécules entre en «colision». L'enchaînement des réactions obtenues par ce système permet de simuler la transduction de signaux.

5.4.2.4 L'interconnexion et l'extension de MAS

Des systèmes multi-agent développés indépendamment, éventuellement à l'aide de technologies différentes, peuvent être interconnectés si leurs protocoles de communication respectent certains standards. De plus, le paradigme multi-agent décrit toute entité logicielle autonome et capable de communiquer avec ses paires comme un agent. Nous avons vu que les services web sont capables de communiquer entre eux et montrent de plus en plus d'autonomie. Un MAS peut donc reposer sur des services web. **`Agmial`** [Bryson00b] est une évolution de **`GeneWeaver`** visant à être hautement distribuée et dont les agents sont implémentés sous la forme de services web, joignables

à la fois par les autres agents et par des utilisateurs *via* des pages web. Un système multi-agent peut également être étendu pour utiliser des services web externes, sous réserve d'unifier leurs protocoles de communication (*via* des agents traducteurs par exemple).

5.4.3 L'implémentation et l'organisation du MAS de `P.A.R.A.DIS.E`

5.4.3.1 Les choix techniques

Pour l'implémentation de notre système multi-agent, nous avions le choix entre de nombreuses librairies compatibles avec le langage **`Java`**. On citera notamment **`Actor Foundry`** [actorfoundary], **`Cougaar : Cognitive Agent Architecture`** [cougaar], **`JADE`** [jade], **`Jason`** [jason] et **`OpenCybele`** [opencybele].
Nous avons choisi d'utiliser la librairie **`JADE`** (**J**ava **A**gent **De**velopment Framework), développée conjointement par *Italia Telecom* et l'Université de Parme. Notre choix s'est porté sur cette librairie pour plusieurs raisons :

- elle répond totalement aux standards de la *FIPA* (Foundation for Intelligent Physical Agents)
- elle est distribuée sous licence GPL (GNU Public Licence), ce qui nous assure de pouvoir l'utiliser dans notre infrastructure sans enfreindre de licence.
- son développement est assuré par un grand groupe industriel et un groupe universitaire en plus de la communauté *Open Source*
- son développement ayant débuté en 1998, elle est l'une des librairies les plus avancées à ce jour
- son interface de programmation (API) est de qualité et une grande source de documentation est disponible
- sa communauté de développeurs et d'utilisateurs est très active

Grâce à **JADE**, il est possible de décrire les agents et leurs comportements à l'aide du langage **Java**, et nous permet donc d'exploiter les avantages de ce langage. Par exemple, l'héritage et la composition nous permettent de définir des agents généraux qui seront ensuite spécialisés en fonction de nos besoins. La communication entre les agents de **JADE** repose sur **Java RMI** et permet d'échaner des objets complexes. **JADE** a notamment été utilisé en bioinformatique pour développer un MAS hiérarchique dédié à la simulation de réseaux biologiques [Shafaei05].

5.4.3.2 L'organisation du MAS de P.A.R.A.DIS.E

Le système multi-agent de **P.A.R.A.DIS.E** devra contenir les agents responsables des différentes ressources (algorithmes et interfaces graphiques). A son démarrage, l'environnement contenant les agents est vide, et les agents le rejoigne au besoin. Comme nous l'avons vu, nous souhaitons que l'architecture **P.A.R.A.DIS.E** soit distribuée, afin de permettre l'utilisation des algorithmes sur un serveur au sein d'un laboratoire et celle des interfaces graphiques sur l'ordinateur de chaque utilisateur.
L'organisation choisie pour le déploiement de **P.A.R.A.DIS.E** est donc de démarrer le MAS sur un serveur et d'y intégrer un agent par algorithme disponible. Dans **P.A.R.A.DIS.E**, la structure composée du cœur du MAS et des agents responsables d'algorithmes est appelée la *plateforme*. Chaque interface graphique de **P.A.R.A.DIS.E** est associée à un agent qui rejoint la plateforme quand l'outil graphique est démarré.

Au démarrage de **P.A.R.A.DIS.E** un écran d'accueil propose à l'utilisateur de démarrer la plateforme et/ou les outils graphiques (figure **5.7**). Le démarrage de la plateforme affiche une fenêtre de gestion des agents (figure **5.8**), permettant de suivre en temps réel le statut des différents agents connectés et les échanges de messages entre ces agents.

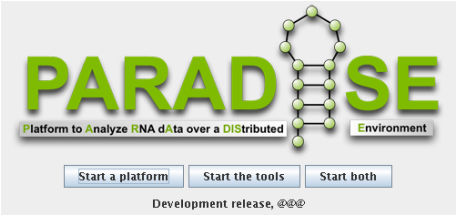


FIG. 5.7 – L'écran d'accueil de **P.A.R.A.DIS.E**

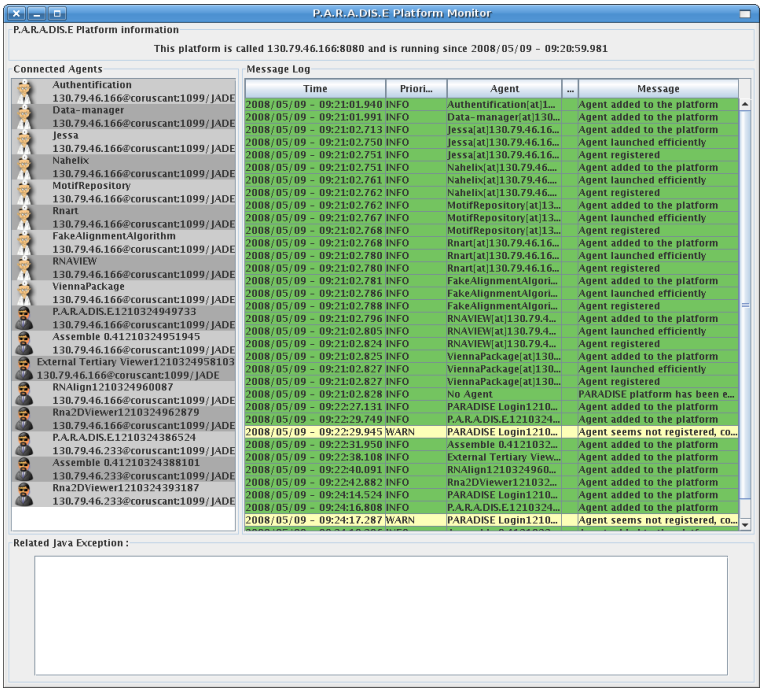


FIG. 5.8 – Le gestionnaire de plateforme

Cette interface graphique permet de surveiller l'activité d'une plateforme ***P.A.R.A.DIS.E***

Une fois la plateforme déployée, il est possible de démarrer, sur un ordinateur du réseau, les interfaces graphiques et de rattacher leurs agents à la plateforme. Ceci provoque l'affichage d'une fenêtre graphique (figure **5.9**) demandant à l'utilisateur de fournir l'adresse de la plateforme à contacter (l'adresse réseau de l'ordinateur hébergeant la

plateforme).

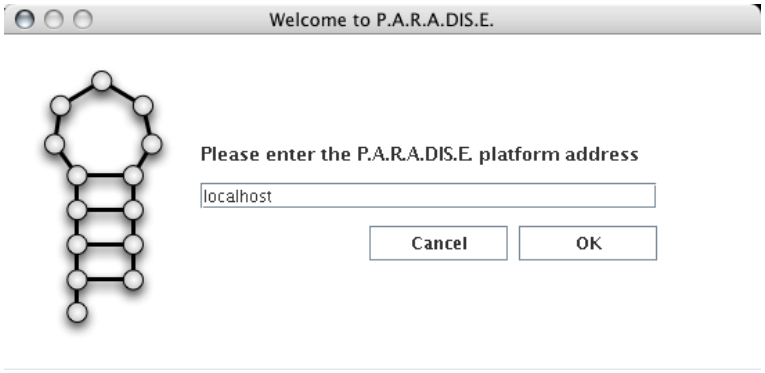


FIG. 5.9 – La fenêtre de connexion
Cette fenêtre demande l'adresse de la plateforme à contacter

Une fois l'adresse fournie, la *barre d'outil* de **P.A.R.A.DIS.E** (figure **5.10**) s'affiche et propose un bouton pour chaque type d'interfaces graphiques disponible (voir **5.2**). La séléection d'un de ces boutons démarre l'interface graphique associée, ainsi que son agent qui rejoint automatiquement la plateforme spécifiée au démarrage.

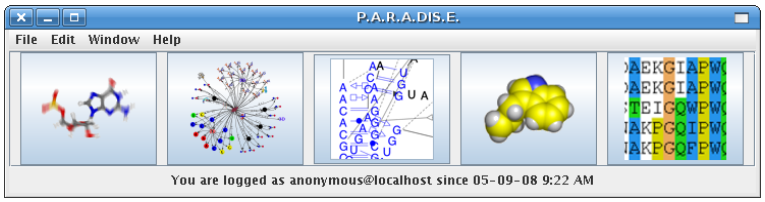


FIG. 5.10 – La barre d'outils de **P.A.R.A.DIS.E**
Cette barre d'outil permet d'ouvrir différents formats de fichiers (PDB, CT, FASTA, RNAML, BPSEQ), et propose un bouton pour lancer chacun des outils graphiques disponibles dans ***P.A.R.A.DIS.E***

5.4.4 Les agents de `P.A.R.A.DIS.E`

Les agents de **`P.A.R.A.DIS.E`** sont chacun en charge d'une ressource particulière et sont divisés en deux groupes en fonction du type de cette ressource.

5.4.4.1 Les agents responsables d'interfaces graphiques

Un agent responsable d'un outil graphique (*ParadiseTool*) est appelé un `ParadiseToolAgent`. Il a pour rôle de permettre à l'utilisateur d'accéder aux algorithmes susceptibles de fournir des données interprétables par l'interface graphique dont il est responsable.
Lorsqu'un utilisateur souhaite contacter un algorithme, l'interface graphique transforme les actions de l'utilisateur en une requête (voir **5.4.5.1**). Cette requête est alors transmise à l'agent de l'interface graphique qui est chargé de trouver l'agent responsable de l'algorithme et de lui soumettre cette requête (voir **5.4.6**).
Le `ParadiseToolAgent` a également pour but de contacter l'interface graphique, à la demande d'autres agents. Ainsi, lorsqu'un résultat d'algorithme est proposé au `ParadiseToolAgent`, celui-ci extrait les objets du modèle de **`P.A.R.A.DIS.E`**, contenus dans le message et les transmet à l'interface graphique. Celle-ci va alors synchroniser ces objets avec ceux déjà présents. L'affichage graphique est alors remis à jour pour tenir compte de ces nouvelles informations. L'agent a également pour rôle de prévenir l'interface graphique lorsqu'il reçoit un message d'erreur de la part d'un autre agent (le plus souvent pour indiquer que l'algorithme n'a pas pu produire de résultat). Ce message d'erreur est alors affiché par l'interface graphique pour prévenir l'utilisateur. Enfin, l'agent est chargé de transmettre à son interface graphique d'éventuels messages d'information en provenance d'autres agents. Le but de ces messages est le plus souvent d'informer de l'état d'avancement de longs algorithmes. Ainsi, l'interface graphique peut mettre à jour la barre de progression correspondante à la requête en cours, afin que l'utilisateur se rende compte de son évolution.

5.4.4.2 Les agents responsables d'algorithmes

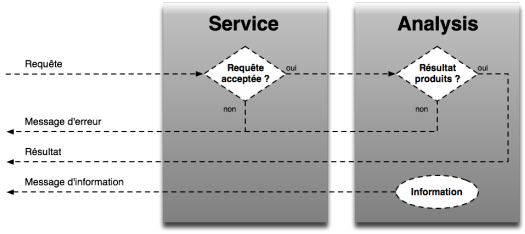


FIG. 5.11 – Le comportement des *Analysis* et *ParadiseService*

Dans **P.A.R.A.DIS.E**, les algorithmes sont pilotés par des agents spécialisés : les *ParadiseService*. A la réception d'une requête d'un autre agent, le *ParadiseService* décide de l'accepter ou non. Si la requête est acceptée, une analyse (*Analysis*) est créée et menée jusqu'à la production d'un résultat final à la requête ou jusqu'à ce qu'une erreur survienne. En plus du résultat et d'éventuels messages d'erreur, le *ParadiseService* est également chargée de transmettre à l'agent à l'origine de la requête des messages d'information produits par l'*Analysis* pendant le déroulement de l'algorithme. Le comportement des *ParadiseService* est schématisé sur la figure **5.11**.

5.4.5 Les requêtes

5.4.5.1 Le format des requêtes

Comme nous l'avons vu, les messages envoyés par les *ParadiseToolAgent* aux *ParadiseService* pour l'exécution d'un algorithme contiennent des requêtes. Ces requêtes permettent d'ajuster l'activité de l'algorithme à l'aide de trois types d'informations :

1. les molécules qui doivent subir l'analyse par l'algorithme
2. les *Feature* de ces molécules à prendre en compte durant l'analyse
3. une série de paramètre, propre à chaque algorithme et permettant de l'ajuster

Les molécules sont à fournir par la requête car, si elles existent sur l'ordinateur de l'utilisateur, elles sont absentes de la plateforme qui contient les algorithmes. Les `Feature` sont à fournir afin de lever toute ambiguïté. En effet, un algorithme peut effectuer une analyse sur une structure secondaire, mais la molécule contenue dans la requête peut posséder plusieurs structures secondaires alternatives, il est donc nécessaire de préciser quels `Feature` sont ciblés par chaque analyse. Ce sont les molécules qui contiennent la liste des `Feature` sélectionnés pour être analysés.

5.4.5.2 Les rôles des *`Analysis`*

Quand le `ParadiseService` reçoit un message contenant une requête, elle transmet celle-ci à l'`Analysis` dont elle est responsable. Le rôle de l'`Analysis` est alors de vérifier la conformité de cette requête. Un algorithme attend un certain nombre de données en entrée (par exemple : n molécules, une structure tertiaire pour chaque molécule, m paramètres). Si certaines informations sont manquantes, l'`Analysis` va envoyer un message d'erreur à l'agent à l'origine de la requête pour l'en informer.

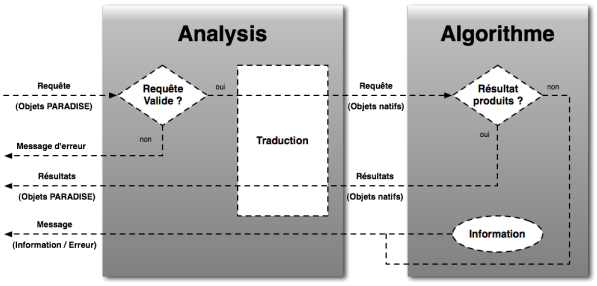


FIG. 5.12 – Le comportement des `Analysis` et algorithmes

Dans le cas contraire, son rôle est de transformer ces données, actuellement sous la forme d'objets **`P.A.R.A.DIS.E`**, en données interprétables par l'algorithme et d'ordonner son exécution. L'`Analysis` joue donc un rôle de traducteur entre **`P.A.R.A.DIS.E`** et l'algorithme, ce rôle est également primordial pour transformer le résultat de l'algorithme en objets **`P.A.R.A.DIS.E`**.

Enfin, le dernier but de l'`Analysis` est de transformer tout message (d'information ou d'erreur) qu'elle reçoit de l'algorithme en message transmis par son

`ParadiseService` au destinataire idoine. Le comportement des `Analysis` est représenté sur la figure **5.12**.

5.4.6 L'identification des agents

Lorsqu'un `ParadiseToolAgent` doit soumettre une requête à un algorithme, il doit d'abord identifier le `ParadiseService` responsable de cet algorithme au sein de la plateforme. Pour ce faire, un agent particulier, provenant de l'infrastructure **JADE**, et fournissant un service de *pages jaunes*, est présent dans la plateforme. A leur arrivée dans la plateforme, les différents `ParadiseService` entrent en contact avec l'agent de pages jaunes et s'enregistrent auprès de lui. Lorsqu'un agent est à la recherche d'un agent fournissant un service (algorithme) particulier (prédiction de repliement en structure secondaire, production d'un dessin de structure secondaire, ...), celui-ci s'adresse donc à l'agent de pages jaunes qui lui indique l'identifiant de `ParadiseService` à contacter.

5.4.7 La communication entre les agents

La communication entre les agents de **P.A.R.A.DIS.E** (schématisée sur la figure **5.13**) repose sur sept catégories de messages, décrits par la norme **ACL** de la FIPA (Foundation for Intelligent Physical Agents) [fipa].

- `Subscribe` : un agent souhaitant soumettre une requête à un `ParadiseService`, initie la conversation par ce type de message. Ceci lui permet de s'assurer que le `ParadiseService` cible est bien disponible.
- `Confirm` : un `ParadiseService` recevant un message `Subscribe` notifie sa disponibilité par un message `Confirm`. Ce système évite donc à un agent de soumettre une requête contenant potentiellement des nombreux objets, et donc étant lourde en ressources réseau, à un `ParadiseService` indisponible.

- `Request` : ce type de message est utilisé par un agent pour soumettre une requête à un `ParadiseService`. Ce message contient donc les différentes informations décrites en **5.4.5.1**.
- `Propose` : Lorsqu'un résultat est produit par un algorithme, son `ParadiseService` le transmet à l'aide d'un message `Propose` à l'agent à l'origine de la requête
- `Inform` : un agent envoie ce type de message à un autre pour lui transmettre une information
- `Failure` : ce type de message est transmis par un `ParadiseService` pour informer qu'aucun résultat n'a pu être produit par l'algorithme à la requête soumise. Ce type de message peut être déclenché si la requête est mal formatée (voir **5.4.5.1**) ou si l'algorithme est intrinsèquement incapable de produire un résultat à partir des données dont il dispose (c'est à dire qu'il n'existe pas de réponse à la question soumise à l'algorithme). Ce type de messages peut également être déclenché à la suite d'un problème de nature informatique (problème de lecture de fichier, capacité de mémoire dépassée, «*bug*»,...). Le message **`Failure`** contient toujours la raison de son émission et dans le dernier cas, il contient également un journal d'erreur «*Stack Trace*» permettant aux développeurs d'identifier précisement l'origine du problème.
- `Cancel` : un agent envoie ce type de message pour mettre fin à une conversation et donc annuler les calculs et requêtes en cours. Ce type de message est envoyé à la suite d'une action de l'utilisateur.

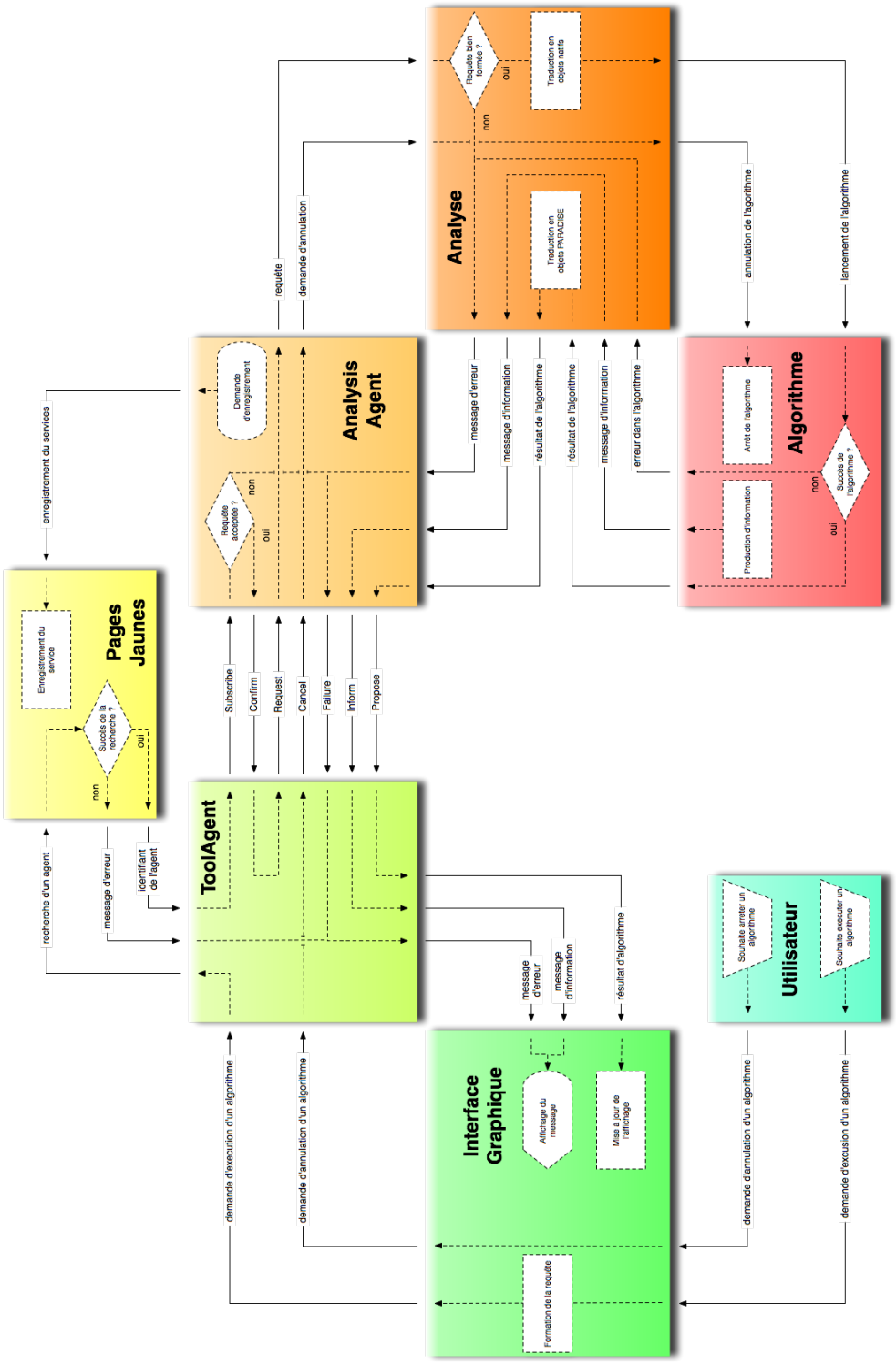


FIG. 5.13 – La communication entre les modules de **P.A.R.A.DIS.E**

Ce diagramme schématise le système de communication de ***P.A.R.A.DIS.E****. Les flèches à traits pleins représentent des messages échangés entre différents modules. Les flèches à traits pointillés représentent les réaction de chaque module aux messages reçus.*

5.4.8 Les algorithmes de `P.A.R.A.DIS.E`

La mise en place de notre infrastructure nécessite la disponibilité d'algorithmes pour un certain nombre de fonctionnalités. Dans un premier temps, nous avons décidé de fournir un algorithme unique, le plus représentatif à nos yeux de chaque type de fonctionnalité, et de nous focaliser sur les algorithmes choisis afin de tester l'efficacité du système et de mettre en place d'éventuelles améliorations. La figure **5.14** reprend le diagramme de la figure **5.1** en nommant les outils graphiques et algorithmes de la plateforme **`P.A.R.A.DIS.E`**.

5.4.8.1 RNAView

L'algorithme **`RNAView`** [Yang03] permet d'obtenir une structure secondaire étendue annotant une structure tertiaire. L'utilisateur fournit un fichier au format **`PDB`** et **`RNAView`** détermine automatiquement la structure secondaire des molécules contenues dans ce fichier et l'exportera dans un fichier au format XML respectant la grammaire **`RnaML`** [Waugh02]. Cette structure secondaire comprend les hélices (intramoléculaires et intermoléculaires), ainsi que la liste de toutes les paires de bases présentes en accord avec la nomenclature *Leontis-Westhof*. En plus de la structure secondaire, **`RNAView`** fournit un dessin de celle-ci, obtenu par projection orthogonale des coordonnées des nucléotides sur un plan. La structure secondaire étendue obtenue par **`RNAView`** peut être étudiée à l'aide de **`RNA2DViewer`** et servir de structure de référence dans l'alignement de la molécule avec des orthologues dans **`RNAlign`**.

5.4.8.2 RNAdistance

`RNAdistance` [Hofacker94] du **`RNA Vienna Package`** permet de calculer une distance entre structures secondaires. Cet algorithme est utilisé pour produire un premier jet d'alignement.

5.4.8.3 RNAfold

L'algorithme **RNAfold** [Hofacker03] du **RNA Vienna Package** utilise une approche de minimisation de l'énergie libre afin de produire une structure secondaire à partir d'une séquence proposée. Cette structure secondaire ne pourra toutefois contenir que des paires de bases canoniques et wobble. Cet algorithme est utilisé :

- pour obtenir une structure secondaire à partie d'une séquence non repliée
- pour préaligner une séquence contre la molécule de référence dans **RNAlign**. La structre secondaire obtenue et la molécule de référence sont alignées à l'aide de **RNAdistance**

Le couplage de ces deux outils ne permet pas forcement d'obtenir le meilleur alignement possible, mais celui-ci constitue un bon départ pour un alignement manuel ultérieur. De plus, contrairement aux approches visant à trouver une structure secondaire consensus, il assure que lorsque la structure secondaire d'une séquence est connue, celle-ci ne sera pas modifiée par l'algorithme d'alignement.

5.4.8.4 RNAplot

RNAplot utilise les informations de structure secondaire pour produire un dessin de cette structure. Les coordonnées des bases issues de ce dessin pourront être utilisées par l'outil **RNA2DViewer**. Le dessin proposé par cet algorithme a la propriété de présenter un recouvrement minimal entre les éléments de la structure secondaire, ce qui assure une grande lisibilité.

P.A.R.A.DIS.E propose également trois autres algorithmes étroitement liés au développement de notre outil **Assemble**.

1. **Nahelix**, qui est chargé de fournir un premier jet de structure tertiaire à partir d'une structure secondaire

2. **`Motif Repository`**, permettant d'appliquer un repliement tridimensionnel particulier sur une partie d'une structure tertiaire
3. **`RnaRT`**, qui permet d'affiner les coordonnées d'une structure tridimensionnelle en fonction de contraintes structurales

Ces algorithmes sont donc détaillés dans le chapitre **7**.

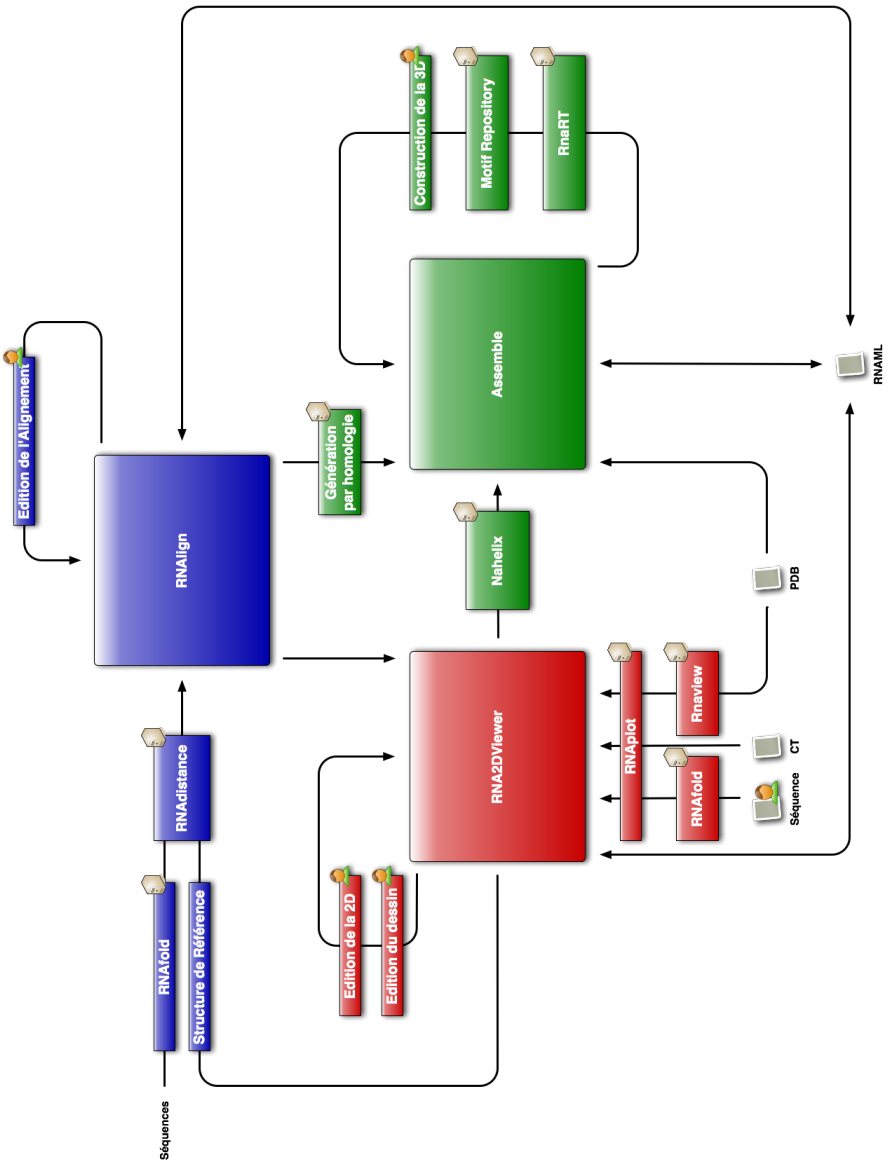


FIG. 5.14 – Les outils et algorithmes de **P.A.R.A.DIS.E**

Ce diagramme montre les successions d'analyses possibles avec ***P.A.R.A.DIS.E****. En rouge : la partie structure secondaire, en bleu : la partie alignement structural et en vert la partie structure tertiaire.*

5.5 Les entrées et sorties de l'infrastructure P.A.R.A.DIS.E

P.A.R.A.DIS.E peut utiliser plusieurs formats de fichiers pour charger des données dans une session de travail ou pour les sauvegarder : le format **CT** pour les structures secondaires, **PDB** pour les structures tridimensionnelles et **Fasta** pour les alignements et jeux de séquences non annotées. Toutefois, le format le plus adapté à la lecture/écriture des données de **P.A.R.A.DIS.E** est le format **RnaML** (*RNA Markup Language*) [Waugh02]. Ce format, qui suit la norme XML, permet de stocker des informations de séquence, structure secondaire, structure tertiaire, alignement et interaction intermoléculaire, tout en intégrant la nomenclature *Leontis-Westhof*. **P.A.R.A.DIS.E** est capable de lire des fichiers **RnaML** dans sa version actuelle (1.0). Toutefois, cette version officielle présente à nos yeux quelques défauts. Nous avons donc entrepris de développer une version plus adaptée à nos besoins. En effet, la version officielle de **RnaML** nécessite la présence de nombreuses balises XML imbriquées pour le stockage d'informations, même très simples. Par exemple, la position de la base en 5' d'une hélice est une information numérique qui nécessite l'imbrication de trois éléments XML alors qu'elle pourrait simplement être stocké sous la forme d'un attribut XML. Cette complexité rend les fichiers très volumineux, notamment lorsque ceux-ci contiennent des structures secondaires et tertiaires de molécules de grandes tailles. En plus d'entraîner le gaspillage d'espace sur le disque stockant ces fichiers, qui peuvent vite être assez nombreux, le désavantage majeur de cette grande taille réside dans le temps requis pour lire et écrire un fichier. Les simplifications syntaxiques que nous proposons permettent de réduire environ de moitié la taille des fichiers. La figure **5.15** montre la différence entre les deux versions de **RnaML** en ce qui concerne la sauvegarde des coordonnées 3D et la description des hélices et paires de bases.

```
<base>
  <position>1</position>
  <base-type>U</base-type>
  <atom serial="6">
    <atom-type> P</atom-type>
    <coordinates>-15.1 227.4 -126.1</coordinates>
  </atom>
  <atom serial="17">
    <atom-type> O3'</atom-type>
    <coordinates>-11.1 225.1 -127.7</coordinates>
  </atom>
</base>

<helix id="H79">
      <base-id-5p>
            <base-id>
                  <molecule-id ref="1"/>
                  <position>974</position>
            </base-id>
      </base-id-5p>
      <base-id-3p>
            <base-id>
                  <molecule-id ref="2"/>
                  <position>82</position>
            </base-id>
      </base-id-3p>
      <length>10</length>
</helix>

<base-pair comment="?">
      <base-id-5p>
            <base-id>
                  <molecule-id ref="1"/>
                  <position>2</position>
            </base-id>
      </base-id-5p>
      <base-id-3p>
            <base-id>
                  <molecule-id ref="2"/>
                  <position>21</position>
            </base-id>
      </base-id-3p>
      <edge-5p>W</edge-5p>
      <edge-3p>S</edge-3p>
      <bond-orientation>T</bond-orientation>
</base-pair>
```

```
<base base-id="1" molecule-id="0">
  <atom
    type="P"
    x="-15.1" y="227.4" z="-126.1"
  />
  <atom
    type="O3'"
    x="-11.1" y="225.1" z="-127.7"
  />
</base>


<helix
      id="H79"
      molecule1-id="1"
      base5-id="974"
      molecule2-id="2"
      base3-id="82"
      length="10"
/>


<base-pair
      molecule1-id="1"
      base1-id="6"
      edge1="W"
      molecule2-id="2"
      base2-id="21"
      edge2="S"
      orientation="T"
/>
```

FIG. 5.15 – Comparaison entre les deux version de **RnaML**

*A gauche la version 1.0 de **RnaML**, à droite la version utilisée dans **P.A.R.A.DIS.E**. De haut en bas, des blocs décrivant les coordonnées 3D des atomes d'une base, une hélice et une paire de bases.*
On peut constater la présence d'informations redondantes dans la section dédiée aux coordonnées 3D (le type de base peut être connu en consultant la séquence) et des balises XML inutiles dans la description des hélices et paires de bases.
Le fichier en version 1.0 contient 774 caractères, alors que celui dans notre version en contient 332.

Troisième partie

La modélisation moléculaire d'ARN

Chapitre 6

Introduction

La fonction d'un ARN au sein de la cellule est déterminée par son architecture. Le décryptage des règles qui dirigent le repliement des architectures d'ARN est donc essentiel à la compréhension de la fonction de ces molécules et à l'analyse de leur évolution. La compréhension de ces règles nécessite toutefois la disponibilité d'un grand nombre de structures tridimensionnelles d'ARN. Bien que nous disposions à l'heure actuelle d'un très grand nombre de séquences d'ARN, l'obtention de structures de haute résolution par cristallographie de rayon X ou résonance magnétique nucléaire reste très difficile. Ceci est d'autant plus vrai que la taille des molécules d'ARN étudiées est importante. Il en résulte un écart entre le nombre de molécules séquencées et celles pour lesquelles une structure tridimensionnelle a été déterminée.

Afin de réduire cet écart, la modélisation moléculaire est une alternative pour la production de structures tertiaires d'ARN lorsque les approches expérimentales se montrent infructueuses. La modélisation moléculaire est un ensemble de méthodes théoriques et de techniques informatiques permettant de modéliser des molécules et de simuler leur comportement. Les techniques employées reposent notamment sur la chimie informatique et consistent à affecter des coordonnées cartésiennes à chaque atome de la molécule considérée afin d'obtenir une structure tridimensionnelle.

Le principe général de la modélisation moléculaire est de décomposer les structures déjà résolues expérimentalement en blocs de construction, et de les assembler ensuite, afin de construire un modèle pour la molécule considérée, en accord avec les règles structurales connues. Suivant l'approches considérée, la taille des blocs de construction s'échelonnent de deux ou trois nucléotides à des domaines structuraux complets.
La modélisation moléculaire s'effectue en quatre étapes :

1. analyse de la molécule à modéliser, et identification des éléments la constituant
2. création de blocs de construction correspondant à ces éléments, soit en les extrayant de structures résolues expérimentalement, soit en les générant algorithmiquement
3. assemblage de ces blocs pour former le modèle moléculaire
4. analyse et correction du modèle obtenu

Deux types d'approches, illustrées par des publications récentes seront détaillées dans ce chapitre :

1. les approches automatiques, qui sont dirigées par un algorithme
2. les approches semi-automatiques, dans lesquelles l'utilisateur supervise la modélisation

6.1 Les approches automatiques

Au cours de l'année passée, deux programmes permettant la détermination de la structure tertiaire de molécules d'ARN ont été publiés. Ils se reposent sur la décomposition automatique de structures en entités structurales élémentaires et leur assemblage par un algorithme.

6.1.1 FARNA

FARNA [Das07] est un programme qui vise à prédire la structure tertiaire d'une séquence en utilisant un échantillonage de fragments élémentaires évalué par une fonction d'énergie personnalisée. Comme le squelette sucre-phosphate de l'ARN est articulé par sept angles de torsion, une approche visant à trouver la meilleure structure possible en faisant varier ces angles est trop complexe pour être réalisable. Pour éviter ce problème, **FARNA** utilise une méthode de Monte Carlo (voir chapitre **3**) pour former le squelette sucre phosphate en assemblant des fragments (trinucléotides) issues de la structure cristallographique de la grande sous-unité du ribosome de *H.marismortui*. Ces fragments sont choisis en fonction de la nature des nucléotides (purine ou pyrimidine) à la position considérée. Un grand nombre de structures sont ainsi générées et classées à l'aide d'une fonction d'énergie prenant en compte :

- le rayon de rotation en chaque point, favorisant un repliement hélical en conformation de type A
- les interférences stériques, évitant aux nucléotides d'entrer en collision
- la possibilité de former des paires en fonction de la distance entre les bases
- la possibilité de former des paires en fonction de la coplanarité entre les bases
- l'empilement des bases

Ce programme a été testé sur un jeu de 20 molécules dont la structure est connue. Dans environ la moitié des cas, l'une des cinq «meilleures» structures proposées présente une RMSD inférieure à 4 Å avec le squelette de la structure résolue. Les paires de bases Watson-Crick et non Watson-Crick sont prédites avec une efficacité respective de 90% et 35%.

6.1.2 MC-Sym

L'approche de **MC-Sym** [Parisien08] est analogue à celle de **FARNA** dans le sens où le programme cherche à reconstituer une molécule d'ARN en assemblant des pièces

élémentaires. Toutefois, au lieu de chercher à assembler des fragments de nucléotides, l'approche de **MC−Sym** repose sur la juxtaposition de cycles. Ainsi, l'ensemble des structures de la NDB sont décomposés en cycles d'interactions, c'est à dire en groupes indivisibles de nucléotides reliés par des appariements et empilements (figure **6.1**). La notion de cycle d'interactions, introduite dans [Lemieux06], est également utilisée pour rechercher toutes les séquences compatibles avec un motif d'ARN donné [St-Onge07].

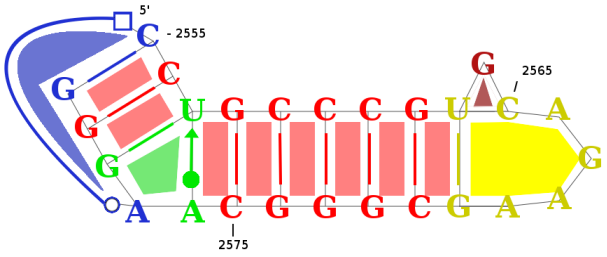


FIG. 6.1 – Les cycles d'interactions

La structure secondaire étendue de l'hélice 2555-2580 de la grande sous-unité du ribosome de H.marismortui annotée par sa décomposition en cycles minimaux d'interactions.

Lorsque l'utilisateur fournit une structure secondaire, indiquant les appariements Watson-Crick et non Watson-Crick et les empilements de bases, **MC−Sym** décompose celle-ci en cycles d'interactions et recherche dans sa banque de cycles issus de structures tridimensionnelles ceux qui sont le plus adaptés au regard de la séquence, puis les assemble.

6.1.3 Les limitations

Ces deux approches souffrent de limitations communes :

– elles font totalement abstraction du contexte biologique de la molécule (données d'expérimentations biochimiques ou biophysiques, partenaires connus,...)

– en travaillant à une échelle très petite (trois ou quatre nucléotides), elles garantissent que la structure soit localement correcte (car correspondant à une portion de structure observée), mais il n'y a aucune garantie de l'exactitude globale. En effet, il existe plusieurs candidats valables dans la banque pour la plupart des fragments/cycles, et le choix entre ces candidats peut influer sur le résultat global. Par exemple, le remplacement d'un fragment/cycle candidat par un autre à une position charnière peut modifier l'aspect général de la molécule en rapprochant ou éloignant deux bras de la structure.

Ces limitations font que, plus une molécule est grande, plus le changement d'échelle entre le niveau local (fragment/cycle) et le niveau global est important, et de fait, au delà d'une certaine taille de molécule (30 et 150 nucléotides pour **FARNA** et **MC−SYM** respectivement), les performances chutes drastiquement. Or, ce sont les molécules de grande taille qui sont les plus difficiles à cristalliser et qui, par conséquent, nécessitent le plus d'outils de prédiction efficaces.

Les limitations de FARNA

L'approche de **FARNA** présente un défaut supplémentaire : elle est dirigée par la construction du squelette, et les paires de bases sont formées par effet de bord. Il semble plus adapté de chercher d'abord à former des paires de bases qui auront ensuite pour effet de stabiliser le squelette. Ces failles dans l'approche de **FARNA** sont vraisemblablement à l'origine des principales faiblesses du programme :

– le fait que les paires de bases ne soient pas prédites avec une efficacité de 100%
– la difficulté à prédire des bulles internes

Ce programme obtient des résultats tout à fait honorables pour des molécules de très petite taille, et ces performances peuvent surement être améliorées en optimisant les pondérations de la fonction d'énergie. Toutefois, des molécules de cette taille ne peuvent pas adopter une structure d'une complexité extrême. En effet, des éléments structuraux tels que des jonctions entre trois ou quatre hélices ou des interactions à longues distances

ne peuvent pas être observés dans des structures de cette taille. Et si le programme est capable de prédire des structures en tige boucle pour des petites molécules, il est compréhensible qu'il lui soit plus difficile de prédire des repliements complexes pour des molécules de grande taille.
Une méthode efficace pour améliorer les performances de cet algorithme (en tout cas pour la prédiction de structures de petite taille) serait de limiter son espace de recherche en permettant à l'utilisateur de fournir une série de contraintes structurales issues d'expérimentations humides (par exemple une liste de bases impliquées ou non dans la formation de paires, comme c'est le cas dans des logiciels comme **MFOLD** [Zuker03]). Comme de nombreux programmes, **FARNA** gère les appariements entre deux bases, et non les interactions impliquant d'autres atomes, comme le groupement hydroxyle en 2' du ribose. L'intégration de tels critères dans la fonction d'énergie peut vraisemblablement augmenter les performances pour certaines structures. Enfin, si l'utilisation de l'échantillonnage permet de faire efficacement abstraction des angles de torsion le long du squelette, l'importance de l'angle χ entre le ribose et la base ne devrait pas être négligée. Introduire un certain degré de liberté sur cet angle permettrait à certains fragments de former des paires de bases ou des empilements qu'ils ne peuvent former dans leur état original et ainsi d'obtenir un meilleur score en satisfaisant plus de contraintes.

Les limitations de MC-Sym

Le prédiction de structure tertiaire par **MC-Sym** consiste à décomposer une structure secondaire en cycles et à trouver un candidat pour représenter chaque cycle en trois dimensions. Toutefois, toutes les structures secondaires ne disposent pas de suffisamment d'informations pour être décomposées en cycles, et une prédiction de structure secondaire par **MC-Fold** peut être nécessaire. Cette prédiction a pour avantage de tenir compte des interactions non Watson-Crick, mais ajoute également un certain degré d'incertitude. En effet, si la reconstruction en trois dimensions se base sur

une structure secondaire erronée, elle ne peut être correcte. De plus, cette méthode à un temps d'exécuter très long pour des arns de plus que quelques dizaines de nucléotides, tant dans l'étape de prédiction de structure secondaire que dans la recherche de cycles candidats.

6.2 Les approches semi-automatiques

Les approches présentées dans la section précédente ont pour but d'assembler des fragments de très petites tailles, sélectionnés indépendamment, afin de produire un modèle moléculaire et sont non supervisées par l'utilisateur.
Les approches présentées dans cette section laissent à l'utilisateur le soin de construire lui-même le modèle de la molécule tout en automatisant le plus d'opérations possibles. L'idée générale de ce type d'approches est de construire le repliement du modèle en suivant le même cheminement que la molécule elle-même. En effet, la molécule nécessite plusieurs étapes avant d'acquérir sa conformation fonctionnelle. D'abord, des empilements et appariement entre les nucléotides se forment et constituent les hélices définissant la structure secondaire de la molécule. La compaction de ces éléments de structure secondaire dans un espace à trois dimensions permet de former les interactions tertiaires et la structure tridimensionnelle. Le repliement de la molécule est donc hiérarchique.
Cette hiérarchie guide les approches interactives de modélisation moléculaire. La structure secondaire permet de définir un certain nombre d'éléments structuraux (hélices et simples brins), ces éléments sont générés en 3D et assemblés de manière à former les interactions tertiaires souhaitées.

6.2.1 RNA2D3D

RNA2D3D [Martinez08] est l'un des très rares programmes permettant de faire de la modélisation moléculaire d'ARN de façon interactive. Ce programme propose, à partir d'une structure secondaire, une première approximation d'un modèle tridimensionnel supposé être perfectionné par l'utilisateur. Pour ce faire, le programme produit un dessin en deux dimensions de cette structure qui est ensuite projeté en trois dimensions. Les nucléotides en 3D sont donc répartis sur un plan et les hélices sont alors embobinées pour adopter une conformation de type A. A ce stade, partout où cela est possible, les hélices sont prolongées de sorte que leurs nucléotides en 5' et 3' forment des pseudo-paires de bases. Cette étape est appelée la compaction. Ensuite, les hélices voisines sont empilées afin de former une structure énergétiquement plus stable. Les brins d'ARN qui ne sont pas impliqués dans des paires de bases ou pseudo-paires de bases peuvent alors subir un repliement automatique. Pour ce faire, la conformation d'un fragment d'ARN disponible dans une autre structure peut être appliquée sur le brin souhaité, à condition que les séquences soient compatibles. Enfin, une étape d'affinement de coordonnées à l'aide d'un champ de force est effectuée grâce aux outils de la suite logicielle **Tinker** [tinker] (voir **7.6**).
RNA2D3D propose diverses options, comme la possibilité d'empiler automatiquement des hélices, d'ajouter ou supprimer des appariements entre nucléotides ou de faire de la superposition globale ou locale entre deux structures.

6.2.2 Les approches du laboratoire

Des modèles moléculaires d'ARN ont été produits au sein du laboratoire depuis de nombreuses années [Westhof89, Michel90] et le sont toujours actuellement [Beckert08]. Les étapes conduisant à la détermination du modèle moléculaire sont celles décrites dans [Westhof93] et [Masquida05] :

– détermination de la structure secondaire par expérimentations humides et/ou analyse comparative de séquences
– identification de domaines structuraux : hélices et motifs structuraux d'ARN (voir **2.6**)
– génération de ces domaines en trois dimensions
– assemblage de ces domaines
– affinement des coordonnées du modèle

Afin de réaliser ces différentes étapes, plusieurs outils logiciels, dont un récapitulatif est donné en figure **6.2**, ont dû être développés au sein du laboratoire.

6.2.2.1 NAHELIX

NAHELIX [Westhof90] est un programme en ligne de commande permettant de générer la structure tertiaire d'une double hélice d'acides nucléiques en fonction de sa séquence. Ce programme est capable de générer des hélices d'ARN ou d'ADN de différentes conformations (A, B, Z, triple hélice, ...). La génération d'un simple brin est également possible (en donnant une séquence vide pour le second brin de l'hélice). La méthode utilisée pour générer les hélices est détaillée en **7.2.3**.

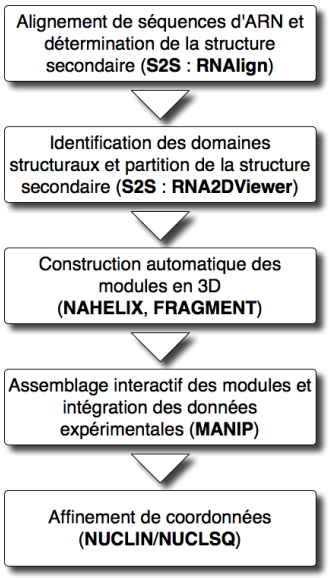


FIG. 6.2 – Les étapes de modélisation avec les logiciels du laboratoire
*Les outils de **S2S** sont décrits dans la section **5.2***

6.2.2.2 FRAGMENT

FRAGMENT [Westhof90] est un outil en ligne de commande permettant de stocker dans un fichier le repliement d'une sélection de nucléotides afin de l'appliquer ultérieurement sur une séquence donnée. L'utilisateur fournit un fichier contenant les coordonnées des nucléotides formant le

motif qu'il souhaite sauver. Les coordonnées des atomes du squelette sucre-phosphate ainsi que l'angle χ entre le sucre et la base sont alors enregistrés dans un fichier de motif. L'utilisateur peut alors fournir une séquence nucléotidique sur laquelle il souhaite appliquer ce motif. Les bases correspondantes à la séquence sont fixés sur les atomes C1' des nucléotides stockés en respectant les angles χ enregistrés et un fichier contenant les coordonnées du «*fragment*» ainsi généré est produit.

6.2.2.3 NUCLIN/NUCLSQ

`NUCLIN/NUCLSQ` [Westhof85] est un programme en ligne de commande inspiré de **`PROLIN/PROLSQ`** [Konnert80] permettant d'affiner les coordonnées d'une structure tridimensionnelle d'ARN en accord avec des contraintes structurales. Cet algorithme, réécrit en **`Java`** sous le nom de **`RnaRT`**, est détaillé en **7.6.3**.

6.2.2.4 MANIP

FIG. 6.3 – La boîte à boutons

Au laboratoire, l'assemblage des modules en trois dimensions se faisait originellement à l'aide du logiciel **`FRODO`** [Jones85]. Le logiciel **`MANIP`** [Massire98] a été développé pour permettre l'assemblage rapide de modules tridimensionnels précédemment générés à l'aide de **`NAHELIX`** et **`FRAGMENT`**, afin de former une architecture complexe d'ARN. Ce programme a été écrit pour être utilisé sur un ordinateur SGI fonctionnant sur le système d'exploitation **`IRIX`**. L'assemblage est effectué interactivement par l'utilisateur à l'aide d'un périphérique SGI appelé *boîte à bouton* (figure **6.3**), possédant plusieurs variateurs permettant de contrôler la translation dans chaque direction et la rotation selon chaque axe des différents modules composant

le modèle moléculaire. Ces variateurs donnent également la possibilité de paramétrer la valeur de chaque angle de torsion des nucléotides, afin d'appliquer un repliement particulier au modèle sans altérer les distances entre les atomes covalents (cette opération est détaillée en **7.4.2**). Afin de permettre à l'utilisateur de manipuler les objets tridimensionnels plus aisément, il est possible d'afficher le modèle de façon stéréographique, ce qui augmente la perception des volumes.

6.2.3 Les limitations

Les limitations de RNA2D3D

RNA2D3D souffre de deux faiblesses dans son approche :

1. Le placement des hélices et la construction du modèle 3D semble trop dirigés par le dessin en 2D de la structure secondaire. Or, l'exemple simple de l'ARN de transfert qui adopte un repliement en feuille de trèfle en 2D et une structure en **L** en 3D montre bien que la manière dont on représente la structure secondaire peut être trompeuse. Certes, dans le cas de l'ARN de transfert l'empilement des hélices corrige rapidement cette première approximation, mais pour des structures aussi complexes qu'un ribosome, cela peut se révéler plus difficile. De plus, la projection de la 2D vers la 3D induit une erreur de distance entre les atomes O3' et P et d'angle entre O3', P et O5'.
2. L'étape de compaction, lors de laquelle les hélices sont prolongées de part et d'autre, semble biologiquement infondée, surtout dans la mesure où elle précède l'empilement de deux hélices. En effet, lorsque des hélices sont prolongées et empilées, les paires de bases directement empilées seront des pseudo-paires de bases, et non les paires de bases décrites dans la structure secondaire. Les simples brins en bout des hélices ne doivent pas être recrutés au sein des hélices, car ils apportent une souplesse et une flexibilité permettant aux hélices de s'empiler (et

induisent parfois des courbures dans les hélices en formant des motifs structuraux comme le Kink-Turn) et de former des jonctions entre trois ou quatre hélices.

De plus, le fait de ne pouvoir appliquer le repliement d'un fragment sur un autre que si leurs séquences sont compatibles est une limitation considérable du processus de modélisation qui est dépassée par **FRAGMENT** grâce à la substitution de bases. La modélisation à proprement dite semble assez rigide dans ce logiciel en comparaison de **MANIP**. En effet, **RNA2D3D** propose de nombreux outils pour travailler à gros grains, en déplaçant des hélices et des sélections, mais des outils permettant une manipulation plus fine, comme la possibilité de modifier les angles de torsion sont totalement absents.

Les limitations de MANIP

Les limitations de **MANIP** sont essentiellement d'ordre technique. Tout d'abord, il n'est possible d'exécuter ce programme que sur un seul type d'ordinateur (SGI), avec un système d'exploitation particulier (IRIX). C'est également le cas des outils «*satellites*» de **MANIP** (**NAHELIX**, **FRAGMENT** et **NUCLIN/NUCLSQ**) développés dans une version non standard de **FORTRAN**. De plus, ces outils ne peuvent être utilisés qu'en ligne de commande, en dehors de **MANIP**. Cela contraint l'utilisateur à faire des aller-retours incessants entre ces différents programmes et à gérer les nombreux fichiers qu'ils génèrent. Enfin, une fois les modules repliés, leur assemblage est entièrement réalisé par l'utilisateur. Ceci est un avantage, car cela confère à l'utilisateur un grand degré de liberté. Cependant, la manipulation et le placement précis d'objets en trois dimensions n'est pas une opération triviale et demande souvent beaucoup de temps. Pour remédier à ce défaut, des automatismes peuvent être développés pour les opérations les plus récurrentes : la superposition et l'empilement d'éléments structuraux.

La chapitre suivant décrit le logiciel **Assemble** développé pour dépasser les limitations de **MANIP**.

Chapitre 7

Le logiciel Assemble

A mon arrivée au laboratoire, j'ai commencé le développement du logiciel **Assemble** (figure **7.1**). Le but de ce travail était d'écrire un outil ayant le même objectif que **MANIP**, tout en dépassant ses limitations et en profitant des dix années d'évolution dans le domaine de l'informatique en générale et de la bioinformatique en particulier. Contrairement à **MANIP** qui ne fonctionne que sur les stations de travail SGI, **Assemble** a été développé pour être utilisable sur n'importe quel type d'ordinateur. De plus, si les grandes étapes permettant la modélisation d'une molécule sont les mêmes entre les deux programmes, **MANIP** nécessite que l'utilisateur fasse appel à des programmes externes en ligne de commande (**NAHELIX**, **FRAGMENT** et **NUCLIN/NUCLSQ**). Ces programmes ont également été réécrits pour être utilisables sur une machine moderne et pour être directement accessibles par **Assemble** *via* l'infrastructure **P.A.R.A.DIS.E**. En guise d'amélioration par rapport à **MANIP**, **Assemble** proposera, de plus, de nombreuses options permettant d'automatiser des opérations d'éditions habituellement effectuées manuellement par l'utilisateur et pouvant s'avérer très coûteuses en temps. Les cartes de densités électroniques représentent un atout majeur pour la construction d'un modèle moléculaire d'ARN. La gestion de celles-ci constitue un grand avantage d'**Assemble** sur **MANIP**. Enfin,

de nombreuses amélioration permettent de rendre l'affichage plus agréable dans **Assemble**.

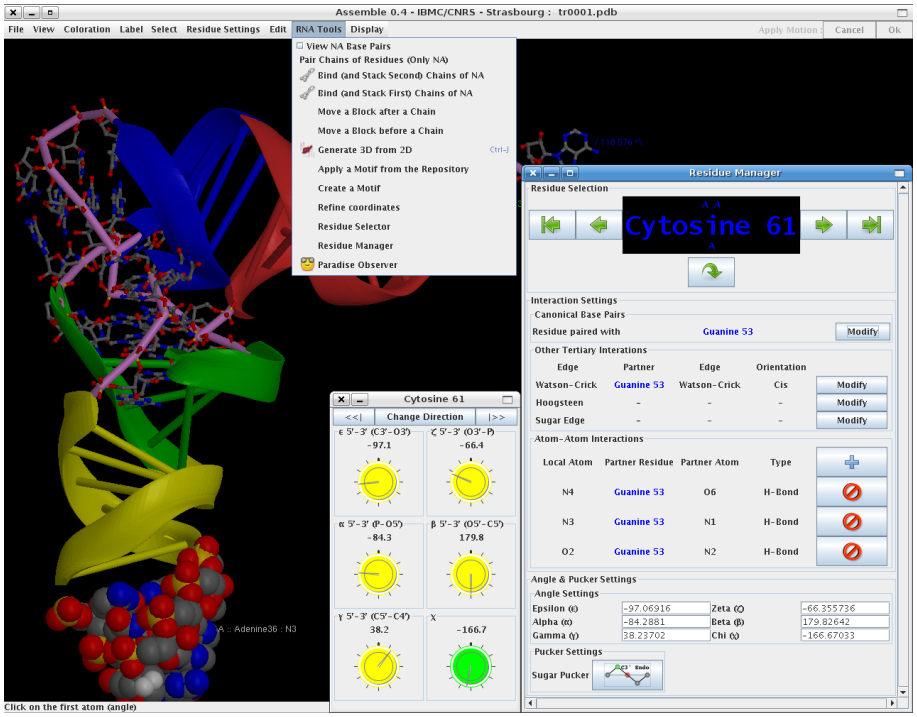


FIG. 7.1 – Assemble

*La fenêtre principale d'**Assemble**, avec de gauche à droite : (1) un modèle d'ARN de transfert, (2) la boîte de torsion, (3) le gestionnaire de nucléotides*

Assemble permet de construire des modèles moléculaires d'ARN, en partant de la structure secondaire, en suivant les quatre grandes étapes déjà disponibles avec **MANIP** :

1. les hélices définies par la structure secondaire sont générées en trois dimensions
2. parmi les simples brins, l'utilisateur cherche à identifier ceux qui peuvent être repliés en accord avec l'un des motifs structuraux disponibles dans un répertoire de motifs
3. ces différents éléments structuraux sont assemblés de façon semi-automatique à l'aide des outils d'édition disponibles
4. le modèle est soumis à une étape d'affinement de coordonnées visant à éliminer les imperfections dues à l'édition manuelle

Le tableau **7.1** indique les différents algorithmes utilisés par **Assemble** *via* l'infrastructure **P.A.R.A.DIS.E**. Les étapes préliminaires permettant la détermination de la structure secondaire, soit par prédiction, soit par alignement sont traitées par les différents outils et analyses de **P.A.R.A.DIS.E**.

Services **P.A.R.A.DIS.E**	Fonction	Se référer à
Jessa	Produire un dessin 2D non recouvrant de la structure secondaire d'un ARN	7.2.2
Nahelix	Générer en 3D un premier jet de la structure tertiaire à partir de la structure secondaire	7.2.3
Motif Repository	Appliquer le repliement d'un motif structural sur une partie du modèle	7.3.1
RnaRT	Procéder à un affinement de coordonnées corrigeant les imperfections du modèle	7.6.3

TAB. 7.1 – Les services **P.A.R.A.DIS.E** utilisés par **Assemble**

7.1 Les choix techniques

Comme pour l'ensemble des outils développés durant les dernières années au laboratoire, **Assemble** a été écrit en **Java**, ce qui le rend utilisable sur tout type de systèmes informatiques. **Assemble** repose donc sur le paradigme de la programmation orientée objet, ce qui le rend à la fois efficace en terme de performances et modulable pour l'ajout de nouvelles fonctionnalités.
Le programme devant être capable d'afficher un nombre parfois très important d'objets (atomes, liens, cartes de densités, textes, ...) de façon plus ou moins complexe en fonction du mode de rendu choisi, il est nécessaire de faire appel à une librairie

permettant de faire traiter toutes les opérations de dessin 3D par la carte graphique.
La norme **OpenGL** [opengl], développé par Silicon Graphics est une librairie de programmation permettant de faire exécuter des opérations par la carte graphique et fonctionnant aussi bien sous **Linux** que **MacOSX** ou **Windows**. Toutefois, étant développé en C et utilisant certaines fonctionnalités propres à chaque système d'exploitation, **OpenGL** n'est pas directement utilisable par une application **Java**. L'utilisation d'une librairie intermédiaire donnant accès, sous forme d'instruction **Java** à ces fonctionnalités écrites en C, est donc nécessaire.
Java 3D [java3d] et **Xith3D** [xith3d] permettent d'accéder à un certain nombre de fonctionnalités **OpenGL** à travers une librairie de concepts objets de hauts niveaux, mais au prix d'une rapidité moindre. La librairie **JOGL** (Java OpenGL) [jogl] permet d'interfacer directement**Java** avec les primitives (fonctions) **OpenGL**, la rendant plus complexe à utiliser, mais plus efficace en terme de rapidité.
Au début du développement d'**Assemble**, **JOGL** était la librairie la plus prometteuse pour l'implémentation de notre application, nous permettant d'accéder à des fonctionnalités de bas niveaux et donc de produire du code le plus efficace possible, et nous avons donc porté notre choix sur cette librairie.

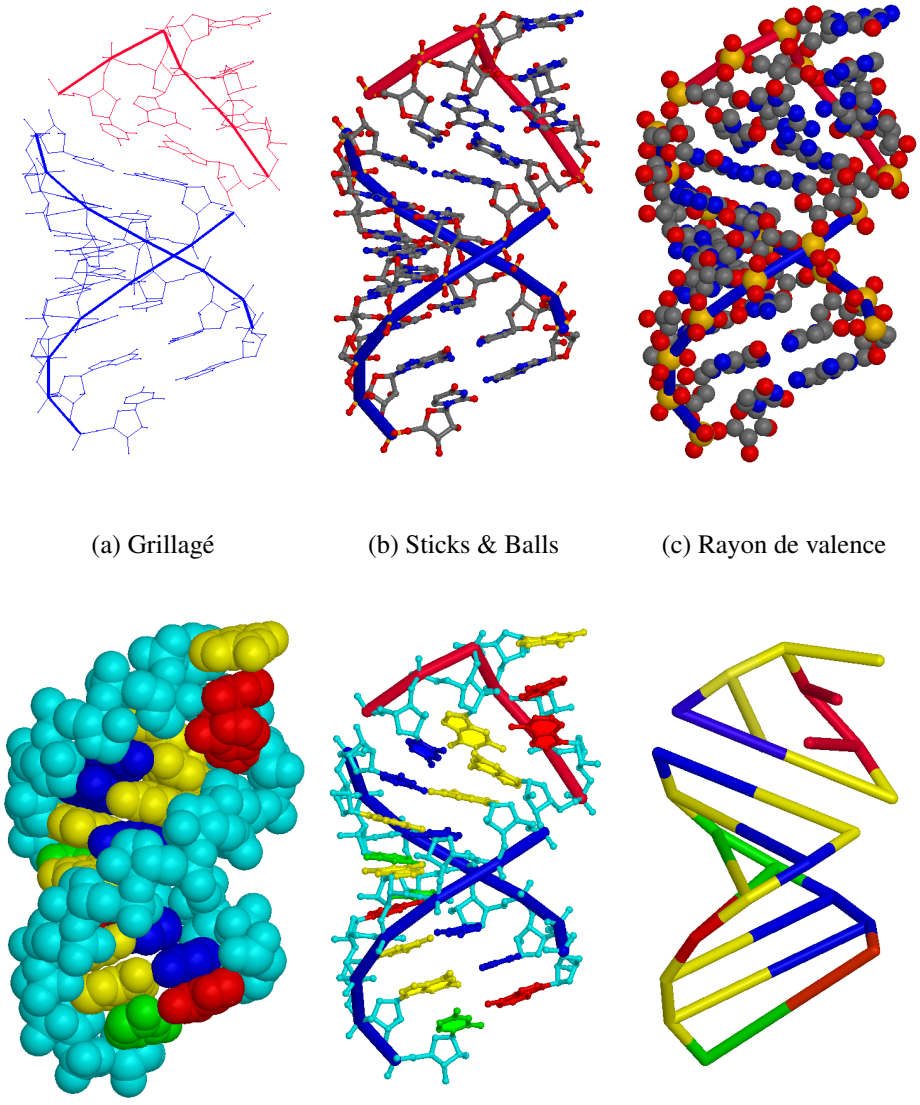
La visualisation du modèle dans `Assemble`

`Assemble` propose un certain nombre de caractéristiques graphiques attendues pour un visualiseur 3D. Il est notamment possible d'afficher l'espace de modélisation (ou scène 3D) en mode *stéréo* et régler finement la distance et l'angle entre les deux scènes pour adapter le rendu à la vue de chacun. L'activation du brouillard (*depth-cueing*) augmente la sensation de profondeur de la scène. De plus, l'utilisateur pourra activer un masquage par plan (*clipping*), pour cacher les éléments plus proches/éloignés qu'une distance donnée.

A sa demande, l'utilisateur pourra afficher sur le modèle des informations telles que :

- le nom d'un atome donné, ainsi que la place du nucléotide dans la séquence
- la mesure d'une distance, d'un angle plan ou d'un angle dihèdre entre des atomes donnés
- la représentation d'interactions entre atomes et une indication de la distance à laquelle ces atomes doivent se trouver l'un de l'autre pour que l'interaction puisse s'établir

Pour aider l'utilisateur à se repérer dans le modèle, **`Assemble`** peut colorer, changer le type de rendu (figure **7.2**) ou masquer n'importe quelle partie de la structure. Cette édition du mode de rendu se fait grâce à une fenêtre graphique adaptée (figure **7.3**). Afin d'aider l'utilisateur à construire son modèle en accord avec le contexte structural de la molécule, d'eventuels partenaires (ARN ou protéines) peuvent être affichées simultanément dans la scène 3D.



(a) Grillagé (b) Sticks & Balls (c) Rayon de valence

(d) Rayon de Van der Waals (e) Plan (f) Rod

FIG. 7.2 – Les différents modes de rendu disponibles dans **`Assemble`**

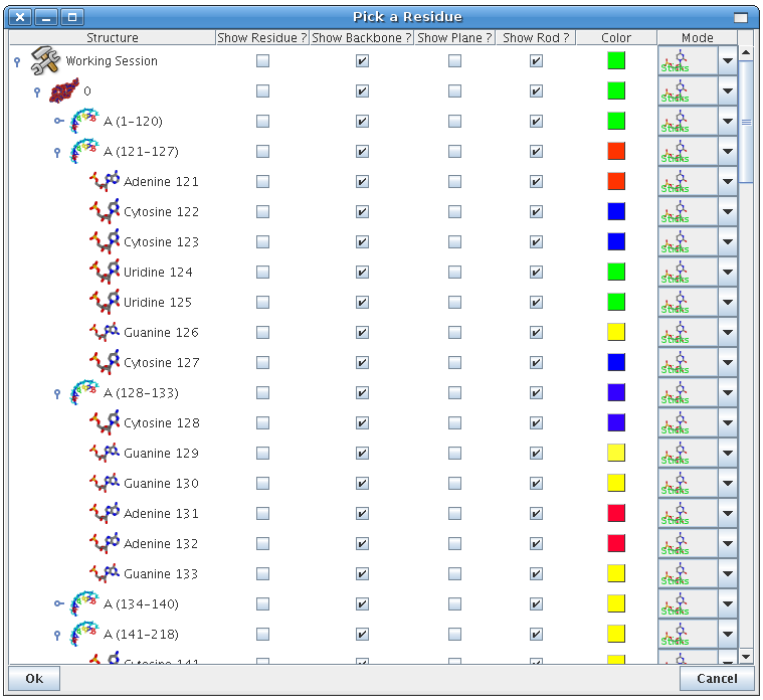


FIG. 7.3 – La gestionnaire de rendus

Cette fenêtre permet de sélectionner des nucléotides, chaînes ou molécules et de choisir leur couleur et mode de rendu.

7.2 De la 2D à la 3D

7.2.1 Pourquoi partir de la structure secondaire ?

La point de départ de la construction d'un modèle moléculaire d'ARN avec **`Assemble`** est la structure secondaire de cette molécule. En effet, comme nous l'avons vu dans le chapitre **2**, la structure secondaire forme le squelette énergétiquement stable de l'architecture d'une molécule d'ARN. Les hélices, qui constituent les éléments de base de la structure secondaire, sont les blocs de construction élémentaires qui, une fois positionnés et assemblés, donnent sa forme au modèle.

7.2.2 La visualisation et l'édition de la structure secondaire

En se reposant sur **P.A.R.A.DIS.E**, **Assemble** a la possibilité d'utiliser une structure secondaire décrite dans un des nombreux formats de fichiers disponibles : **CT**, **RNAML**, **BPSEQ**, **FASTA**, ... Une fois cette structure en mémoire nous voulions donner la possibilité à l'utilisateur de pouvoir l'éditer. Pour ce faire, dans un premier temps, **Assemble** proposait la fenêtre graphique **Jessa** (figure **7.4**). **Jessa** permet à l'utilisateur de visualiser ces structures secondaires sous forme de dessin et à l'aide de la notation parenthésée. L'édition de la structure secondaire en ajoutant ou supprimant des paires de bases et des hélices est également possible. Si l'utilisateur ne dispose pas d'un fichier contenant sa structure, il peut procéder à une construction *de novo*.
Par souci de lisibilité du dessin, nous voulions que le dessin présente le moins de recouvrement possible entre les différents éléments structuraux. Pour ce faire, celui-ci était initialement produit par le service **P.A.R.A.DIS.E** de l'algorithme **Rnasearch**, de la suite logicielle **ESSA** [Chetouani97], que j'avais réécrit en **Java** pour cet usage. Comme cet algorithme n'était pas capable de produire un dessin non recouvrant pour certaines molécules de grande taille, les fonctionnalités d'édition de la structure secondaire ont été transférées à l'outil **RNA2DViewer** (voir **5.2.2.1**) qui fait appel à l'analyse **P.A.R.A.DIS.E** de **RNAplot** du **RNA Vienna Package** pour produire un dessin non recouvrant (voir **5.4.8.4**).

7.2.3 La génération d'un premier jet de structure tertiaire

Lorsque l'utilisateur est satisfait de la structure secondaire dont il dispose, il peut demander la génération d'un premier jet de structure tertiaire reposant sur cette structure secondaire. Cette génération est effectuée par l'algorithme **Nahelix** (figure **7.4**), réécrit en **Java** pour être intégré en tant qu'analyse à la plateforme **P.A.R.A.DIS.E**.

Il possède les caractéristiques suivantes :

- Chaque hélice est générée avec la conformation standard de type A des hélices régulières d'ARN, avec un pas de rotation de 33°, un pas d'élévation de 2.6 Å et un sucre en C3'-*endo*.
- Les simples brins adoptent également cette conformation hélicale et sont empilés en 3' de leurs hélices voisines.

Pour générer le repliement tridimensionnel d'un élément structural (single brin ou hélice), **`Nahelix`** crée le squelette ribose-phosphate de la structure en positionnant des nucléotides de façon régulière. La position et l'orientation d'un nucléotide sont déterminées par applications successives de matrices de transformation sur les coordonnées d'un nucléotide de référence, possédant une structure tridimensionnelle adaptée à la conformation souhaitée de l'hélice. Ceci garantit le respect et la régularité des pas d'élévation et de rotation.

Les bases sont alors fixées à l'atome C1' des nucléotides en accord avec la séquence, avec une conformation «*anti*» (angle χ entre le sucre et la base de 180°) leur permettant de s'empiler et éventuellement de former des paires de bases entre deux brins complémentaires.

Une fois le premier jet de structure tertiaire généré, l'utilisateur peut poursuivre la construction du modèle de deux façons :

- automatiquement, en appliquant à une région sélectionnée le repliement d'un motif structural récurrent (**7.3**)
- manuellement, en déplaçant des parties du modèle et en altérant la valeur des angles de torsion (**7.4**)

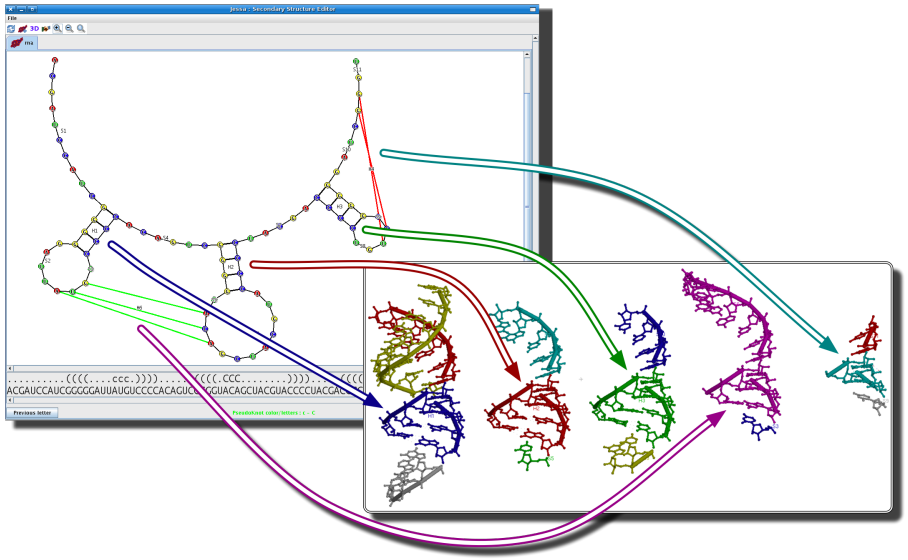


FIG. 7.4 – Jessa & Nahelix

A gauche, la fenêtre **`Jessa`** *permet de visualiser et manipuler les structures secondaires d'ARN. A droite, les hélices et simples brins, décrits par la structure secondaire, tels qu'ils sont générés en 3D par* **`Nahelix`***)*

7.3 L'application de motifs structuraux d'ARN

Comme nous l'avons vu en **2.6**, les réseaux d'interactions constituant l'ARN tendent à former des motifs structuraux. Au même titre que les hélices, ces motifs structuraux sont observés de façon récurrentes à travers les différentes structures disponibles. L'identification d'un motif, au sein d'un domaine, permet donc d'inférer la structure tridimensionnelle de ce dernier. Il est donc important de pouvoir prédire la présence de tels motifs dans la structure que l'on cherche à modéliser. Cette identification se fait en analysant les données dont on dispose (séquence, expériences de biologie humide ou expériences *in silico*, bibliographie, ...). Une fois que l'utilisateur pense avoir identifier un tel motif structural dans sa molécule, **`Assemble`** lui permet d'appliquer le repliement correspondant sur son modèle à l'aide d'un répertoire de motifs structuraux d'ARN.

7.3.1 Le répertoire de motifs structuraux d'ARN

Assemble propose, *via* le service **P.A.R.A.DIS.E** du **Motif Repository**, un répertoire de motifs structuraux d'ARN observés de façon récurrente. Grâce à une fenêtre graphique (figure **7.5**), l'utilisateur peut parcourir ce répertoire et a accès, pour chaque motif disponible, à une série d'informations comprenant notamment :

- son nom
- une description de sa fonction
- une représentation de ses interactions, à l'aide de la nomenclature *Leontis-Westhof*
- une représentation manipulable de son repliement tridimensionnel
- une liste de séquences compatibles avec ce motif
- une liste de publication le concernant
- une liste de structures de la banque **NDB** dans lequelles on peut l'observer
- une liste de partenaires connus

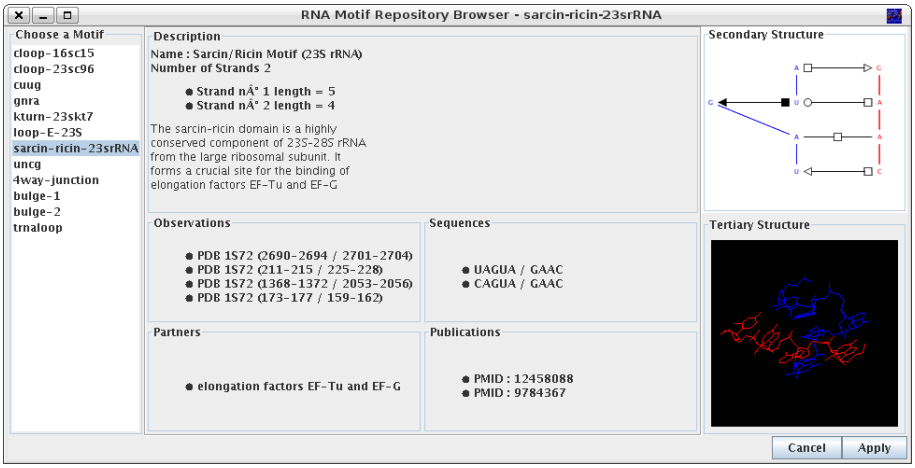
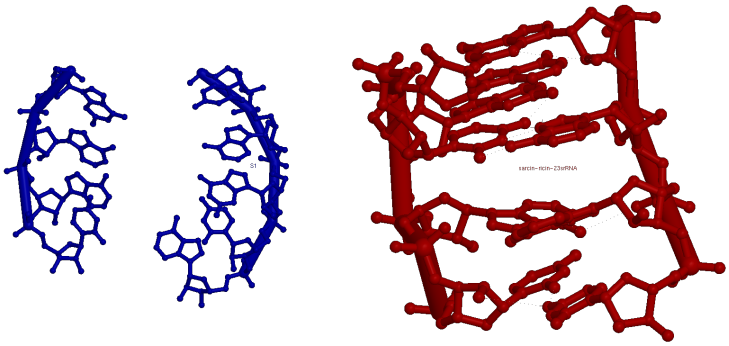


FIG. 7.5 – Le répertoire de motifs structuraux

Cette fenêtre graphique permet de parcourir le répertoire de motifs structuraux d'ARN fourni avec **Assemble** *et affiche diverses informations relatives aux motifs*

7.3.2 Appliquer un motif structural

Une fois que l'utilisateur a fixé son choix sur un motif, il n'a plus qu'à sélectionner la partie de son modèle sur laquelle il souhaite l'appliquer. Cette partie sera alors automatiquement repliée en accord avec le motif et les interactions qui le composent seront affichées sur le modèle moléculaire (figure **7.6**).
Le repliement d'un motif est stocké sous la forme d'une liste de coordonnées pour les atomes du squelette ribose-phosphate et de l'angle χ entre le sucre et la base. L'application du motif sur un domaine de séquence quelconque est possible par substitution de base (la base originale est tronquée et la base correspondante à la séquence est fixée sur l'atome C1' en respectant l'angle χ original).



(a) Avant application (b) Après application

FIG. 7.6 – L'application du motif structural *Sarcin-Ricin*

7.3.3 Etendre le répertoire de motifs

Par défaut, le répertoire fourni avec **`Assemble`** comprend un nombre limité de motifs classiques. Parmi ces motifs, on trouve notamment les boucles GNRA [Jaeger94] ainsi que les motifs Sarcin-Ricin [Leontis02a], Kink-Turn [Klein01], C-Loop

[Lescoute05], ... Cependant, ces motifs d'ARN ne constituent qu'une faible partie des motifs connus, et ceux-ci peuvent exister sous différentes formes. Notre répertoire de motifs ne contient qu'une ou deux versions de chaque motif. L'utilisateur ne trouvera donc pas systématiquement le motif qui l'intéresse sous une forme compatible avec le domaine qu'il souhaite replier dans son modèle. De plus, parfois l'utilisateur souhaite appliquer sur son modèle un repliement qu'il a observé dans une autre structure mais qui ne correspond pas à un motif structural.

Pour pallier à cette faiblesse, **`Assemble`** permettait initialement à l'utilisateur d'ajouter facilement ses propres motifs dans le répertoire. Pour cela, lorsque l'on avait identifié dans une structure (provenant par exemple de la **`NDB`**) un motif que l'on souhaitait ajouter, il suffisait de sélectionner les nucléotides formant ce motif et d'utiliser l'outil de sauvegarde de motif.
Comme l'identification d'un motif d'intérêt se fait très facilement à partir d'une structure secondaire *étendue*, il a été décidé que l'enregistrement de nouveaux motifs se ferait depuis **`RNA2DViewer`** (voir **5.2.2.1**) pour être ensuite utilisé dans **`Assemble`**. Le motif sera désormais disponible dans le répertoire, pour une application ultérieure.
A ce stade du processus de modélisation, l'utilisateur dispose de blocs de construction reflétant en trois dimensions les informations pouvant être tirées de la structure secondaire et d'une analyse de la séquence. Le modèle contient également des brins non repliés (pour lesquels aucun motif structural n'a été identifié). L'étape suivante consiste donc à replier les parties qui ne le sont pas, et à connecter l'ensemble des blocs de construction pour former le modèle moléculaire final. Pour ce faire, **`Assemble`** met à disposition une panoplie d'outils d'édition.

7.4 L'édition manuelle du modèle moléculaire

L'édition du modèle moléculaire dans **`Assemble`** peut se faire à deux niveaux. Soit une approche à gros grain consistant à manipuler des domaines structuraux complets, soit à grain plus fin en éditant directement les nucléotides.

7.4.1 Le déplacement et l'assemblage des blocs de construction

Afin de pouvoir manipuler des domaines structuraux complets, **`Assemble`** introduit la notion de *chaîne* et de *bloc de construction*, dont voici les définitions :

Une ***chaîne*** est une succession de nucléotides contigus et un ***bloc de construction*** est un groupe de chaînes

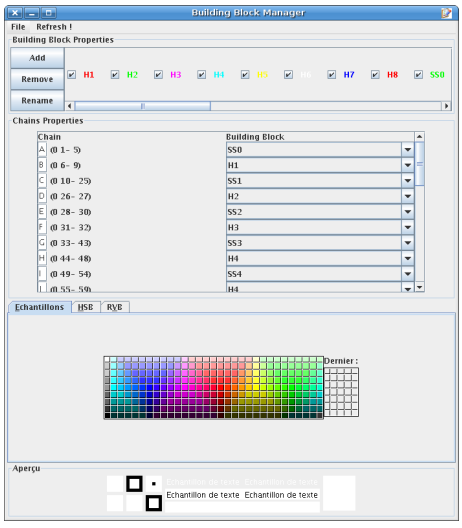


FIG. 7.7 – Le gestionnaire de blocs de construction
Cette fenêtre graphique permet de gérer les blocs de construction et les chaînes

Ces entités peuvent être manipulées de façon indépendante du reste du modèle, et pourront subir des translations dans une direction quelconque ou une rotation autour d'un atome choisi comme pivot.

Par défaut, chaque simple brin, hélice et sélection ayant subit l'application d'un motif structural est stocké dans un *bloc de construction* indépendant. **`Assemble`** propose une fenêtre graphique qui permet à l'utilisateur de gérer les différents blocs ainsi que de les sauvegarder et charger dans un fichier (figure **7.7**).

Afin de pouvoir placer facilement les blocs de construction les uns par rapport aux autres,

Assemble propose des opérations d'édition automatique permettant d'effectuer des empilements et des superpositions de ces entités.

7.4.2 L'édition des angles de torsion

Une fois les domaines structuraux positionnés dans le modèle, il est nécessaire de procéder à l'édition de certains angles de torsion, soit pour fermer une boucle apicale ou une bulle, soit pour lier deux domaines.
Pour replier des brins d'ARN sans en altérer l'intégrité chimique (c'est à dire, sans modifier les distances et angles entre les atomes d'un nucléotide donné), **Assemble** permet de modifier les angles de torsion α, β, γ, ϵ, ζ et χ de chaque nucléotide. Cela peut se faire à l'aide d'une fenêtre graphique dédiée (figure **7.8**), proposant un variateur pour chaque angle, ou bien à l'aide du clavier et de la souris, ce qui permet de ne pas quitter le modèle des yeux pendant l'édition des différents angles.

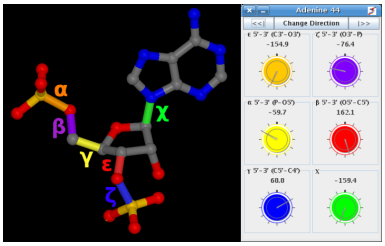


FIG. 7.8 – La boîte de torsion
Cette boîte à boutons permet de modifier les différents angles de torsion du modèle

7.5 La construction du modèle sous contrainte de données expérimentales : les cartes de densité électronique

7.5.1 Introduction

L'ARN étant une molécule très flexible, même en respectant les règles chimiques et géométriques, les possibilités de placement des nucléotides restent très importantes. Il

est donc utile de contraindre d'avantage ce placement et les cartes de densités offrent une information capitale pour y parvenir. Une carte de densité électronique est une information précieuse lorsque l'on cherche à déterminer la structure tridimensionnelle d'une molécule. En indiquant la densité électronique en une série de points d'un espace tridimensionnel, elle permet de prédire, avec plus ou moins de précision, la position des atomes dans cet espace.

Une carte de densité électronique peut être produite par plusieurs procédés expérimentaux, notamment la diffraction de rayons X, la résonance magnétique nucléaire et la cryo-microscopie électronique. Ces expériences permettent de mesurer la densité électronique en des points plus ou moins rapprochés de l'espace, la distance entre ces points définit la *résolution* de la carte.

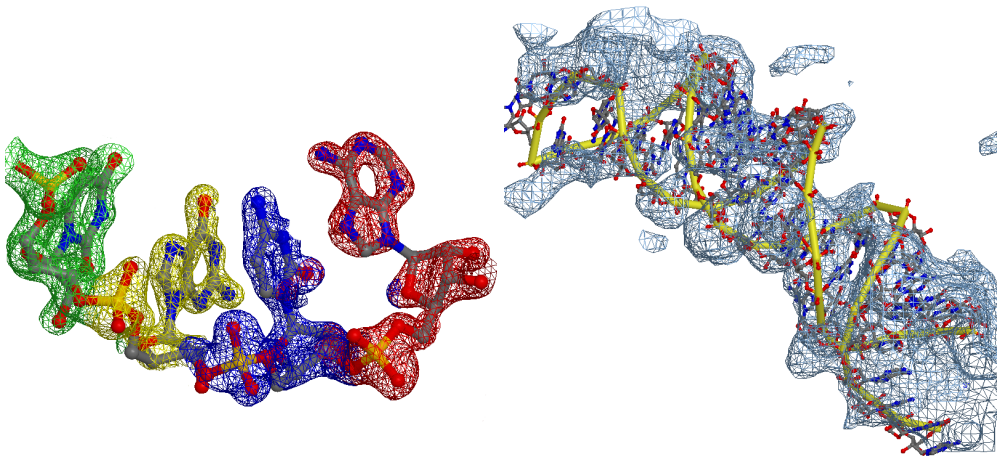
La résolution des cartes peut s'échelonner de moins d'un Ångström, permettant de différencier les types des nucléotides, leurs positions et leurs orientations, jusqu'à 10 Å et au delà permettant tout juste de distinguer la position d'éléments structuraux de grande taille tels que les hélices (figure **7.9**).

7.5.2 La gestion des cartes de densité dans `Assemble`

Une carte de densité électronique permet donc de se rendre compte visuellement de l'aspect général de la molécule, de contraindre et de guider la construction du modèle moléculaire.

`Assemble` permet l'affichage dans la scène 3D d'une ou plusieurs cartes de densité électronique. A l'heure actuelle, **`Assemble`** supporte uniquement le format binaire **`MRC`** et le format `ascii` **`XPLOR`** [Schwieters03], qui peuvent être obtenus par conversion à l'aide de nombreux outils (dont **`mapman`** de la suite logicielle **`CCP4`** [CCP494]).

Quand l'utilisateur charge une carte de densité dans **`Assemble`**, la fenêtre de gestion de carte (figure **7.10**) s'ouvre automatiquement et lui permet de paramétrer l'affichage de la carte (figure **7.11**) en fonction des critères suivants :



(a) Haute Résolution (3.2 Å) (b) Faible Résolution (8 Å)

FIG. 7.9 – Des cartes de densité à haute et basse résolutions

(a) Avec une carte de haute résolution, on peut aisément reconnaître les nucléotides de la chaînes en regardant le maillage de la carte : Uracile en vert, Guanine en jaune, Cytosine en bleu et Adénine en rouge. (b) Avec une carte de basse résolution, il est difficile d'y placer des éléments structuraux, même de grande taille comme cette longue tige boucle.

- sélection du niveau de contour à l'aide d'un histogramme logarithmique
- sélection de la granularité de la carte. Si l'utilisateur choisi une granularité x, la carte affichée ne prend en compte que $1/x$ densité dans chaque dimension. Plus la granularité est faible, plus la carte sera précise, mais plus il faudra de temps pour la calculer. Il peut donc être utile d'augmenter la granularité de la carte lorsque l'on recherche le niveau de contours optimal.
- sélection des coordonnées minimales/maximales d'affichage dans les trois dimensions de la carte
- limitation de l'affichage de la carte à x Å autour d'un atome donné
- choix de la couleur d'affichage de la carte
- choix du type de rendu de la carte entre «maillage», «transparence» et «densité en chaque point»

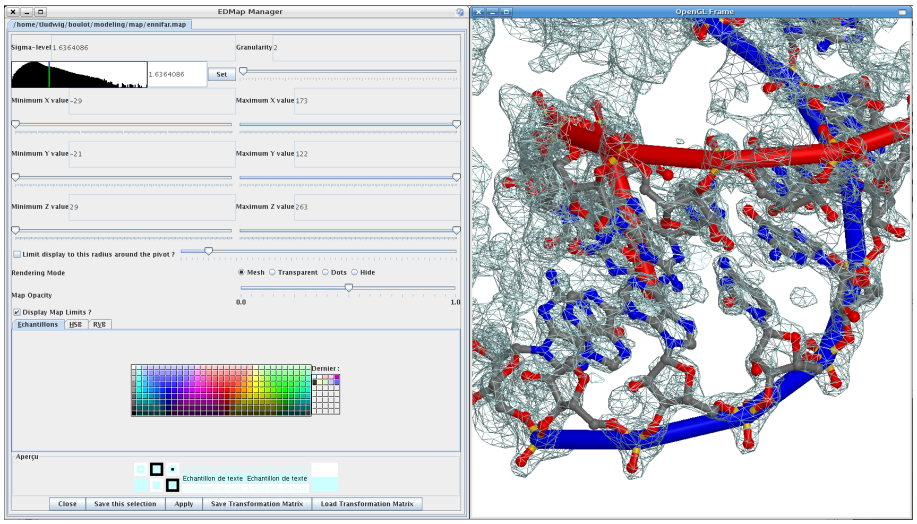


FIG. 7.10 – Le gestionnaire de cartes de densité

Cette fenêtre permet à l'utilisateur de choisir les différents réglages et rendus relatifs aux cartes de densité électronique qu'il a chargées dans **`Assemble`**

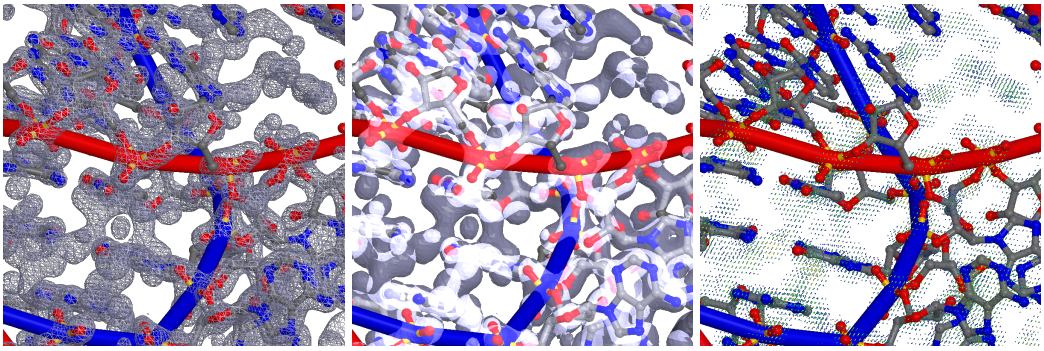
7.5.3 Utiliser plusieurs cartes simultanément

`Assemble` permet l'affichage simultané de plusieurs cartes de densité. L'utilisateur pourra afficher des cartes :

- de résolutions différentes
- pour des domaines différents de la molécule à modéliser
- correspondant à des états différents de la même molécule
- proposant des niveaux de contours différents de la même carte

Lorsque l'utilisateur ouvre plusieurs cartes de densité, il peut les déplacer et les orienter les unes par rapport aux autres afin de les superposer au mieux.

Le chargement d'une carte de densité de taille importante (par exemple une carte de densité d'un ribosome) peut prendre un certain temps. Et si l'on n'est intéressé que par l'un des domaines de la molécule, il peut être frustrant de devoir charger à chaque session la totalité de la carte pour ne travailler que sur quelques nucléotides. C'est pourquoi **`Assemble`** permet de découper et sauvegarder différentes parties d'une carte. Ainsi on pourra avoir, par exemple, une carte de densité par domaine de la molécule à



(a) Rendu grillagé (b) Rendu transparent (c) Rendu par densité

FIG. 7.11 – Les différents rendus de cartes de densités

modéliser. Cette sauvegarde s'effectue en tenant compte des coordonnées des densités, et si l'on ouvre plusieurs cartes obtenues par découpage d'une même carte source, leur affichage les placera dans les coordonnées adéquates.

Une fois la carte de densité affichée, l'utilisateur pourra se servir des différents outils d'édition d'**`Assemble`** afin de placer au mieux les domaines structuraux en densité.

7.6 La correction du modèle final par affinement de coordonnées

7.6.1 Introduction

Étant donné qu'**Assemble** offre à l'utilisateur un grand degré de liberté dans la construction du modèle, celui-ci peut présenter certaines incohérences chimiques et géométriques portant notamment sur les angles et les distances atomiques. **Assemble** propose une étape d'affinement automatique des coordonnées spatiales du modèle afin de minimiser ces erreurs.

7.6.2 Analyse de l'existant

Afin de corriger le modèle, de nombreuses approches reposent sur la mécanique moléculaire. La mécanique moléculaire correspond à l'utilisation de la mécanique newtonienne pour la modélisation de systèmes moléculaires. Dans ce type de systèmes, chaque atome est représenté par une sphère dont le rayon correspond à son rayon de Van der Waals de l'atome. Les liaisons covalentes et autres non-covalentes (ioniques, hydrogène, ...) entre les atomes sont simulées par des «*ressorts*» dont la distance d'équilibre correspond à la distance (théorique ou expérimentalement mesurée) entre les atomes. Un jeu de paramètres et d'équations appelé *champ de force* permettant de décrire l'énergie potentielle du système est utilisé pour faire converger celui-ci vers un point d'équilibre. De nombreuses suites logicielles permettent d'utiliser cette approche (**AMBER** [Case05], **CHARMM** [Brooks83] et **TINKER** [tinker]). Ces outils proposent une minimisation en deux étapes. Dans un premier temps, les atomes trop proches sont repoussés. Puis, une étape de *recuit simulé* «chauffe» la structure pour lui donner de l'énergie et donc de la flexibilité, afin d'explorer un vaste espace de conformations. La structure est alors progressivement refroidie pour parvenir à un état stable satisfaisant

les contraintes. **X-PLOR** et son évolution **CNS** [Brunger07] proposent de minimiser l'énergie du système par la méthode du gradient conjugé (voir **7.6.3**) ou celle du corps rigide. Dans cette dernière, un certain nombre de groupes d'atomes sont définis. Ces groupes sont traités comme des corps rigides et la méthode cherche à minimiser leurs six degrés de liberté de rotation et translation.

Les revues suivantes traitent de l'utilisation de la mécanique molécule, de la dynamique moléculaire et des champs de forces avec des molécules d'ARN [Louise-May96, Auffinger98, Auffinger00, Auffinger07]

7.6.3 RnaRT

Notre approche pour affiner le modèle moléculaire repose sur la satisfaction de contraintes stéréochimiques. Ces contraintes structurales sont :

- la distance entre deux atomes covalents
- la distance entre deux atomes liés par une liaison hydrogène
- l'angle entre deux liaisons covalentes
- la répulsion entre les atomes non liés
- la planarité des bases
- la chiralité des sucres

Les contraintes de distances de covalence, d'angles, de répulsion et de planarité sont automatiquement connues par l'algorithme qui possède un dictionnaire pour ses propriétés pour chaque type de nucléotide. Les contraintes d'interactions entre bases ou entre atomes (liaisons hydrogènes) sont à renseigner par

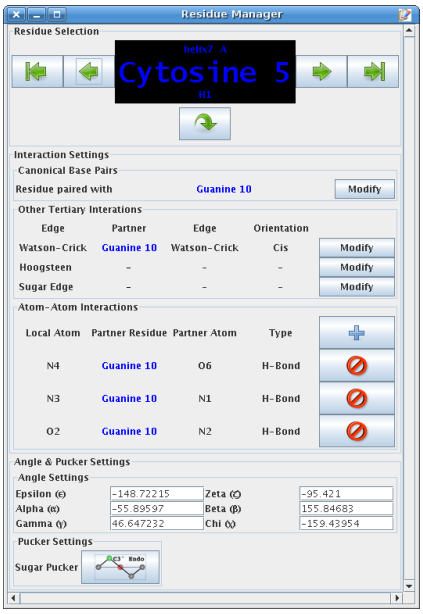


FIG. 7.12 – Le gestionnaire de nucléotides
Cette fenêtre permet à l'utilisateur de choisir les différents contraintes structurales qu'il souhaite appliquer à son modèle

l'utilisateur. Pour ce faire, **`Assemble`** propose une fenêtre graphique permettant de naviguer dans le modèle et de choisir pour chaque nucléotide un partenaire pour chaque face de sa base (figure **7.12**). Il est à noter que la plupart de ces interactions sont pré-remplies par **`Assemble`** quand :

- le nucléotide appartient à une hélice, l'interaction sur la face *Watson-Crick* est renseignée
- le nucléotide a été affecté par l'application d'un motif structural (le motif comprenant en plus des coordonnées atomiques, la liste des interactions qui le composent
- une interaction suivant la nomenclature *Leontis-Westhof* a été définie dans **`RNA2DViewer`**

En utilisant cette fenêtre, l'utilisateur peut également ajuster plus finement les contraintes en ajoutant/supprimant des interactions directement entre les atomes, plutôt que de travailler au niveau des faces. Enfin, la chiralité du sucre d'un nucléotide pourra également être choisie (par défaut tous les ribonucléotides ont une chiralité en *C3' endo*).

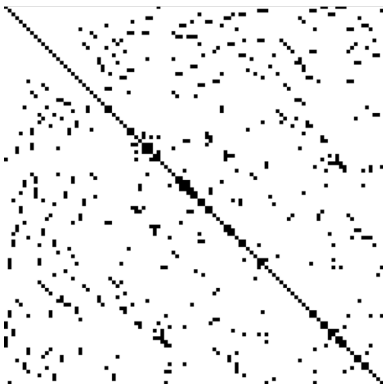


FIG. 7.13 – Matrice creuse
Une matrice creuse est un type de matrice présentant une proportion élevée de valeurs nulles (ici les points noirs sont les valeurs non nulles).

Une fois ces contraintes établies, l'algorithme **`RnaRT`** corrigé le modèle afin que celles-ci soient respectées au maximum. Cet algorithme est une réécriture en **`Java`** de l'algorithme **`Nuclin/NucLSQ`** [Westhof85] précédemment développé en **`Fortran`** au laboratoire. Les distances entre les atomes du modèle sont vues comme des liens élastiques, dont le point d'équilibre est la distance idéale entre les atomes en fonction des contraintes qui les lient. Les différences entre distances observées et idéales pour chaque couple d'atomes sont renseignées uniquement s'il existe une contrainte entre ces atomes. On obtient ainsi une matrice creuse,

symétrique et définie positive (figure **7.13**) contenant les différences de distances. Ainsi, les éléments de la matrice entrent dans deux catégories : les éléments nuls (la grande majorité) et les éléments que l'on va chercher à faire tendre vers 0.

La minimisation de la matrice s'effectue à l'aide de la méthode du gradient conjugé. Il s'agit d'une procédure itérative qui va modifier, à chaque passe, les coordonnées des atomes du modèle et va converger vers un modèle optimal à l'égard des contraintes imposées.

Cette étape est d'autant plus longue que la structure est grande et que le nombre de contraintes imposées est important. Dans l'utilisation, on ne va donc pas chercher à obtenir une structure optimale (c'est à dire une matrice nulle nécessitant de calculer des coordonnées calculées à 10^{-x} Å près), mais un modèle pour lequel, quand une contrainte n'est pas respectée, l'incohérence associée à cette dernière est négligeable.

L'utilisateur va donc fixer l'arrêt de la procédure d'affinement quand :

- un certain nombre de passes a été effectué
- le score de satisfaction de contrainte passe sous un seuil donné
- la taux de progression de ce score entre deux passes successives est inférieur à un seuil donné.

Le gradient conjugé

La méthode du gradient conjugé est un algorithme qui recherche une solution numérique d'un système d'équations linéaires dont la matrice est symétrique et définie positive. C'est une méthode itérative appliquée sur de grands systèmes et est souvent utilisée pour résoudre des problèmes d'optimisation.

Considérons le système suivant :

$$Ax = b,$$

où A est une matrice de taille $n \times n$ symétrique définie positive ($A^\top = A$ et $x^\top Ax > 0$, pour tout vecteur $x \in \mathbf{R}^n$ non nul). Soit $x_\star$ la solution exacte de ce système.

Directions conjuguées

Comme la matrice A est symétrique définie positive, on peut définir le produit scalaire suivant sur $\mathbf{R}^n$: $\langle u, v\rangle_A = u^\top Av$. Deux éléments $u, v \in \mathbf{R}^n$ sont dit A-conjugués si :

$$u^\top Av = 0$$

La méthode du gradient conjugué consiste à construire une suite $(p_k)_{k\in\mathbf{N}^*}$ de n vecteurs A-conjugués. Dès lors, la suite $p_1, p_2, \ldots, p_n$ forme une base de $\mathbf{R}^n$. La solution exacte $x_\star$ peut donc se décomposer comme suit :

$$x_\star = \alpha_1 p_1 + \cdots + \alpha_n p_n$$

où $\alpha_k = \frac{p_k^\top b}{p_k^\top Ap_k}, k = 1, \ldots, n.$

Construction des directions conjuguées

La solution exacte $x_\star$ peut-être également vue comme l'unique minimisant de la fonctionnelle $J(x) = \frac{1}{2}x^\top Ax - b^\top x, \quad x \in \mathbf{R}^n$ On a donc clairement $\nabla J(x) = Ax - b, \quad x \in \mathbf{R}^n$ d'où

$$\nabla J(x_\star) = 0_{\mathbf{R}^n}$$

On définit le résidu du système d'équation comme suit $r_k = b - Ax_k = -\nabla J(x_k)$

r_k représente donc la direction du gradient de la fonctionnelle J en x_k (à un signe près).

La nouvelle direction de descente p_{k+1} suit donc celle du résidu modulo, sa A-conjugaison avec p_k, on a alors :

$$p_{k+1} = r_k - \frac{p_k^\top Ar_k}{p_k^\top Ap_k}p_k$$

 C'est le choix du coefficient $\frac{p_k^\top A r_k}{p_k^\top A p_k}$ qui assure la A-conjugaison des directions p_k. Pour s'en assurer, il suffit de calculer que (Ap_{k+1}, p_k) est bien nul

Algorithme du gradient conjugué

Calcul du résidu $r_0 = b - Ax_0$ pour un vecteur x_0 quelconque. On fixe $p_0 = r_0$.

Algorithme 1 Gradient Conjugé

Pour $k = 0, 1, 2, \ldots$

Faire

$\alpha_k \leftarrow \frac{r_k^\top r_k}{p_k^\top A p_k}$

$x_{k+1} \leftarrow x_k + \alpha_k p_k$

$r_{k+1} \leftarrow r_k - \alpha_k A p_k$

$\beta_k \leftarrow \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}$

$p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$

Fin-Faire

Dans notre cas, on ne cherche pas la solution optimale $x_\star = x_{k+1}$ car elle implique de s'intéresser à des coordonnées calculées avec une précision de 10^{-x} Å ce qui n'est pas réellement significatif au regard de notre problématique. La boucle précédente s'arrêtera donc sous l'une des conditions suivantes :

– k est égale à une valeur donnée choisie par l'utilisateur

– r_{k+1} passe sous un seuil choisi par l'utilisateur

– $\frac{r_{k+1}}{r_k}$ passe sous un seuil choisi par l'utilisateur

Une fois les coordonnées de la structure affinées par **`RnaRT`**, la modélisation de la molécule est achevée. Il appartient à l'utilisateur de vérifier que le modèle obtenu répond bien aux contraintes imposées par la structure secondaire et les différentes informations disponibles concernant la molécule (et en particulier d'éventuelles cartes de densités

électronique). Dans le cas contraire, il faudra retoucher le modèle manuellement et affiner à nouveau ses coordonnées.

FIG. 7.14 – L'affinement de coordonnées

Les tubes jaune et rouge correspondent au squelette sucre-phosphate du modèle avant et après affinement de coordonnées

Chapitre 8

Validation de l'infrastructure : Modélisation d'une molécule d'ARN

Cette partie de la thèse a pour but de mettre en évidence les qualités du logiciel **`Assemble`** à l'aide d'un exemple de modélisation moléculaire. Ce chapitre décrit en détails les différentes étapes de la modélisation de la structure tertiaire d'une molécule d'ARN de 218 nucléotides. La modélisation repose sur deux structures secondaires, obtenues par alignement de séquences et sondages chimique et enzymatique, correspondant à deux états alternatifs de la molécule au sein de la cellule.

8.1 La modélisation du premier état

Le premier état de la molécule est décrit par la figure **8.1**, sa modélisation se déroule comme suit.

8.1.1 L'analyse de la structure secondaire

La première étape, avant de commencer la modélisation de la molécule, est d'identifier les différents éléments structuraux qui la composent. Dans cette structure secondaire, nous avons pu identifier manuellement :

- huit doubles hélices d'ARN (**H1** à **H8**)
- quatre boucles apicales, pouvant se replier en accord avec des conformations disponibles dans notre banque de motifs structuraux (**A** à **D**)
- cinq domaines structuraux complexes, correspondants à des motifs structuraux disponibles dans notre banque (**1** à **5**)

8.1.2 La génération des hélices régulières

Sur la figure **8.1**, on peut voir huit hélices régulières d'ARN. L'hélice **H5** possède quatre bases en bulle (C77, C78, U107 et U108). Nous considérons que ces nucléotides font partie de l'hélice et doivent être générés avec une conformation hélicale. En effet, même si ces nucléotides ne sont pas impliqués dans la formation de paires de bases, dans la structure tridimensionnelle de l'ARN, C77 et C78 seront empilés entre C76 et G79. De la même manière U107 et U108 seront empilés entre U106 et G109.

Une structure secondaire ne décrivant uniquement les appariements des hélices **H1** à **H8** (y compris des paires de bases C77-U108 et C78-U107) est fournie à **`Jessa`** et le premier jet de structure est alors généré par **`Nahelix`**. On procède alors à l'empilement semi-automatique des hélices **H3** et **H4** de manière à ne pas rompre la continuité de la conformation hélice du brins contenant C61 et C62.

8.1.3 La modélisation des boucles apicales

Les boucles **A**, **B** et **C** sont compatibles avec le motif structural GNRA. La boucle **D** est compatible avec le motif UNCG. Ces deux motifs sont disponibles dans notre banque de motifs et sont applicables en quelques clics grâce au **`Motif Repository`**. Pour les boucles **B**, **C** et **D**, on considère que le dernier plateau de bases de l'hélice appartient à la boucle. Ceci est du au fait que les conformations que nous utilisons pour les motifs GNRA et UNCG sont destinés à être appliqués sur des fragments de six nucléotides. Dans ces conformations, le premier et le dernier nucléotide forment une paire de base canonique, ce qui permet d'empiler facilement la boucle sur l'hélice qu'elle termine.
Une fois que les motifs structuraux ont été appliqués, les boucles **A**, **B**, **C** et **D** sont empilées automatiquement en bout des hélices **H4**, **H5**, **H6** et **H8**.

8.1.4 L'application des motifs structuraux

Cinq motifs structuraux disponibles dans notre banque ont pu être identifiés :

1. Le motif Sarcin-Ricin (fig **8.4**). Le repliement tridimensionnel de ce motif, décrit dans [Leontis02a], est issu de la structure cristallographique de la grande sous-unité ribosomale de *H.marismortui* (fichier PDB/NDB : **1S72/RR0082**, nucléotides [2690 ;2694] et [2701 ;2704]).
2. Une jonction entre quatre hélices (fig **8.5**). Le repliement tridimensionnel de ce motif est issu du modèle du snRNA U1 [Krol90] (fichier [U1snRNA], nucléotides [15 ;18], [46 ;49], [90 ;93] et [116 ;119]). Ce type de motif consiste en deux paires d'hélices empilés. Il faut donc choisir quelles hélices seront empilées. Il nous semble plus probable que **H7** soit empilée sur U18-A197, et que **H6** soit empilée sur C19-G118 (la modélisation a été réalisé avec les deux types d'empilement, et seul celui présenté ici est valide, l'autre entrainant des collisions entre différents éléments structuraux).

3. Une jonction entre trois hélices (fig **8.7**). Ce type d'interactions consiste en deux hélices empilées et une troisième hélices pouvant adopter l'une des trois conformations A, B ou C de la figure **8.6**. Les critères exposés dans [Lescoute06b] nous suggèrent un empilement entre **H2** et **H5** et une position perpendiculaire de la dernière hélice (conformation A). Le repliement tridimensionnel de cette jonction, présent dans notre banque de motifs, provient d'un modèle moléculaire et non d'une structure cristallographique.
4. Un nucléotide en bulle (fig **8.8**). Le repliement de ce motif est issu de la structure cristallographique de la petite sous-unité ribosomale de *E.coli* (fichier PDB/NDB **2AVY/RR0123**, nucléotides [1440 ;1442] et [1460 ;1461]).
5. Deux nucléotides en bulle (fig **8.9**). Le repliement de ce motif est issu de la structure cristallographique de la petite sous-unité ribosomale de *E.coli* (fichier PDB/NDB **2AVY/RR0123**, nucléotides [1128 ;1132] et [1142 ;1144]).

Les motifs **4** et **5** ont en fait été ajoutés à notre banque durant la modélisation de cette molécule. En effet, une structure d'ARN peut présenter des bulles et tailles et de séquences très variables et il n'existe pas dans notre banque un motif pour chaque type de bulle. La méthode que nous employons habituellement pour ce type de motif simple est d'analyser la structure secondaire d'une molécule cristallographiée de grande taille, à la recherche d'une bulle candidate ayant la même taille et la séquence la plus proche possible de celle que l'on cherche à modéliser. Après avoir observé ce candidat en 3D pour s'assurer qu'il correspond bien à ce que l'on recherche, le motif est créé grâce à **`RNA2DViewer`** et peut être appliqué sur notre modèle.

Une fois ces motifs appliqués, les différents domaines structuraux sont empilés dans l'ordre suivant :

1. **1** sur **H1**
2. **5** sur **H7**
3. le groupe (**H8**, **D**) sur **5**

4. **4** sur **H2**
5. le groupe (**H3**, **H4**, **A**) sur 4
6. le groupe (**H2**, **4**, **H3**, **H4**, **A**) sur **3**
7. le groupe (**H5**, **B**) sur **3**
8. le groupe (**H6**, **C**) sur **2**
9. le groupe (**H7**, **5**, **H8**, **D**) sur **2**
10. le groupe (**A**, **H4**, **H3**, **4**, **H2**, **3**, **H5**, **B**) sur **2**

8.1.5 La modélisation manuelle

A ce stade, seules les parties en blanc sur la figure **8.1** ne sont pas modélisées.
Les nucléotides A15, C16 et G199 sont modélisés (figure **8.10**) à l'aide des outils d'éditions d'**`Assemble`**, et notamment la **`boîte de torsion`** pour que

- A15 et G199 forment une paire de base trans Hoogsteen Sugar-Edge (appelée *Sheared* et observée fréquement).
- A14, A15, C16 et U17 soient empilés
- A198, G199 et G200 soient empilés

Le fragment [U35 ;C41] est replié à l'aide de la **`boîte de torsion`** pour former une bulle de sept nucléotides entre G34 et G42 (figure **8.11**). Cette étape est la plus délicate de l'ensemble de la modélisation, car elle n'est pas automatisée. De plus, il faut prendre garde que les nucléotides de la bulle n'entrent pas en colision avec l'hélice **H1** ou le motif **1** qui sont très proches des hélices **H3** et **H4** dans le modèle tridimensionnel. C'est d'ailleurs la raison pour laquelle on modélise ce fragment, plutôt que de chercher un motif lui correspondant dans une structure cristallographique.
Les fragments [U1 ;A5] et [A208 ;U218] ne seront pas modélisés, car on ne dispose d'aucune information structurale à leur égard et qu'ils sont vraisemblablement très flexibles.

8.1.6 La correction du modèle

Les empilements automatiques n'étant pas toujours parfaits, il faut en retoucher certains à l'aide des outils d'édition d'**`Assemble`**. Pour finir, une étape d'affinement de coordonnées permettra de corriger les imperfections du modèle et d'obtenir la structure telle qu'elle peut être visualisée sur la figure **8.12**.

La construction de ce modèle n'a demandé qu'une journée de travail. L'essentiel de ce temps a été utilisé pour identifier les motifs structuraux au sein de structures résolues, et pour modéliser le fragment [U35-C41] et la jonction entre **2** et **3**.

8.1.7 Les figures

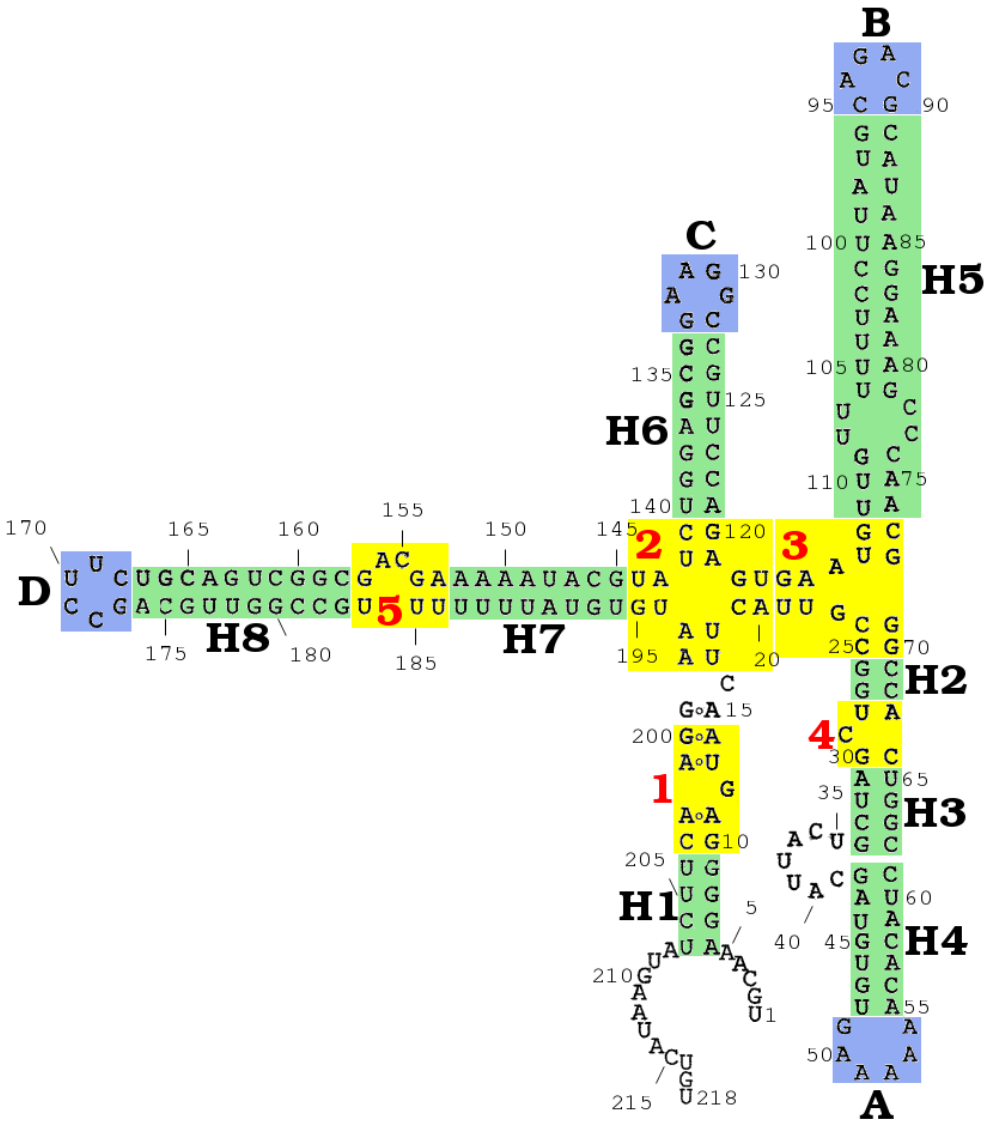
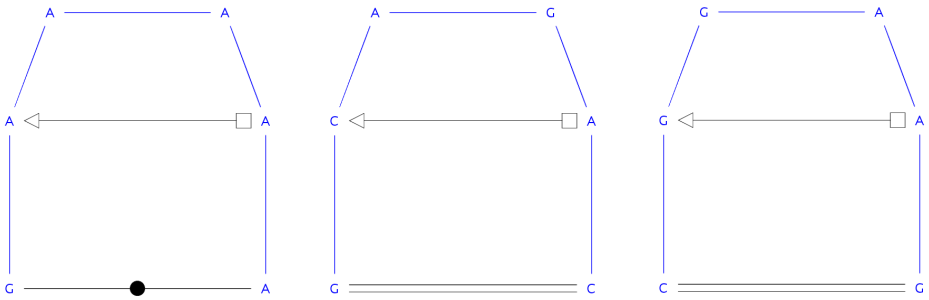
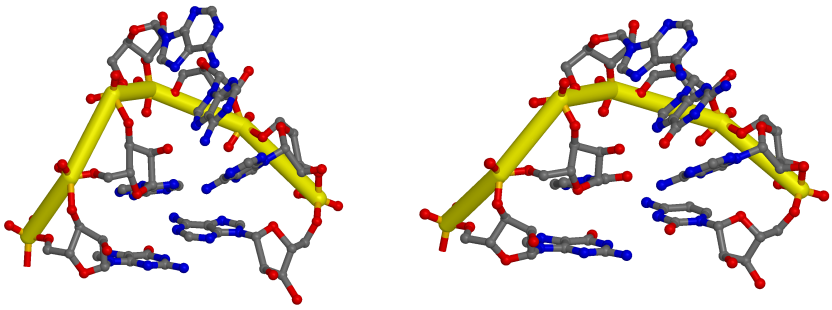


FIG. 8.1 – La structure secondaire du premier état à modéliser

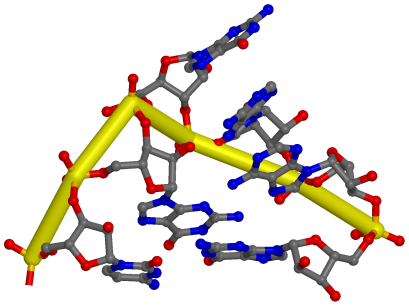
En vert : les doubles hélices d'ARN, en bleu : les boucles apicales et en jaune les motifs structuraux



(a) Boucle **A** (b) Boucle **B** (c) Boucle **C**

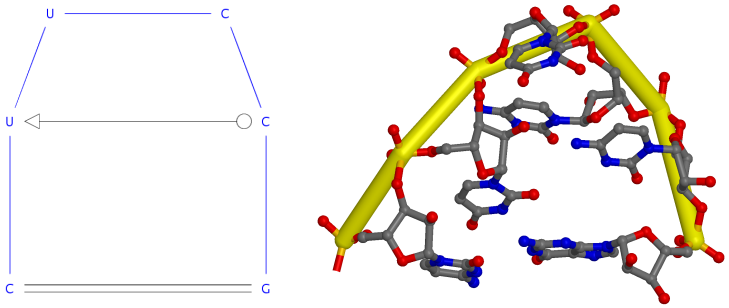


(d) Boucle **A** (e) Boucle **B**



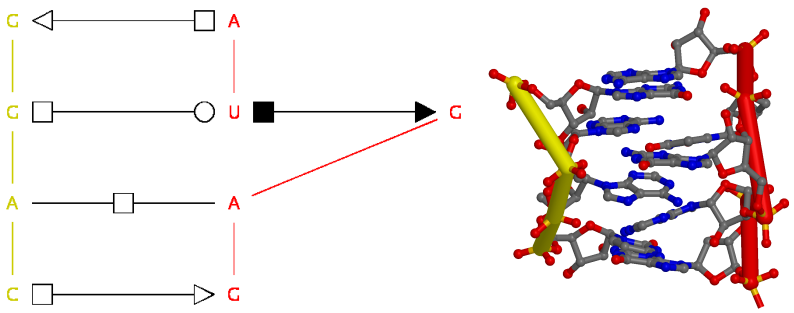
(f) Boucle **C**

FIG. 8.2 – Les boucles **A**, **B** & **C** (GNRA)



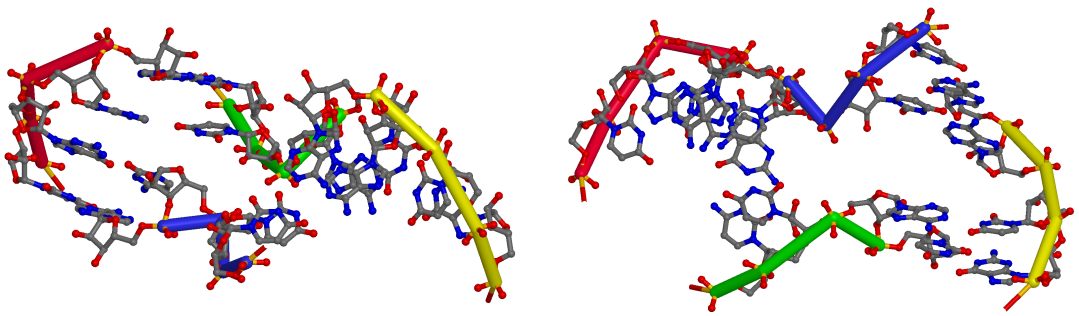
(a) Description (b) Application

FIG. 8.3 – La boucle **D** (UNCG)

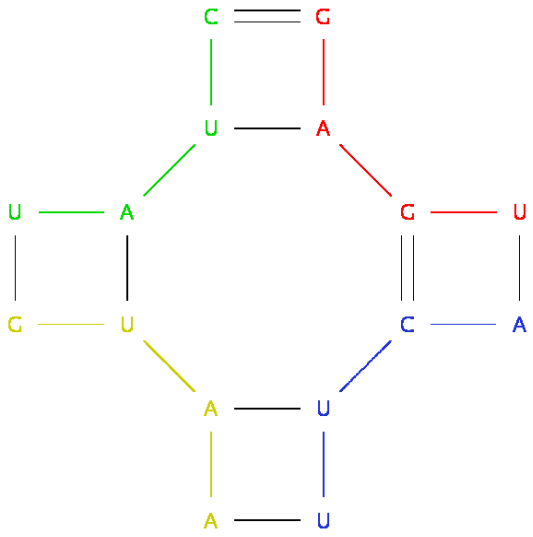


(a) Description (b) Application

FIG. 8.4 – Le motif structural **1** (Sarcin-Ricin)



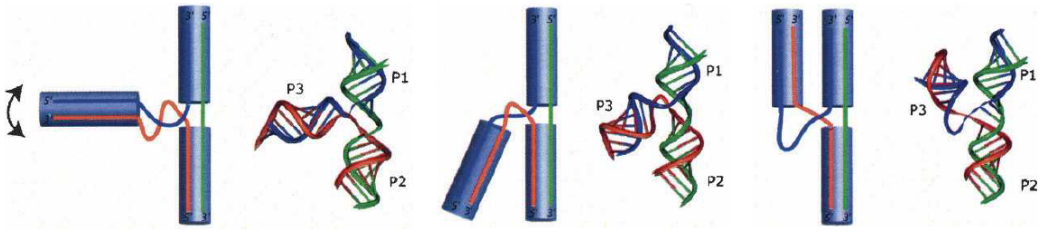
(a) Application (b) Application



(c) Description

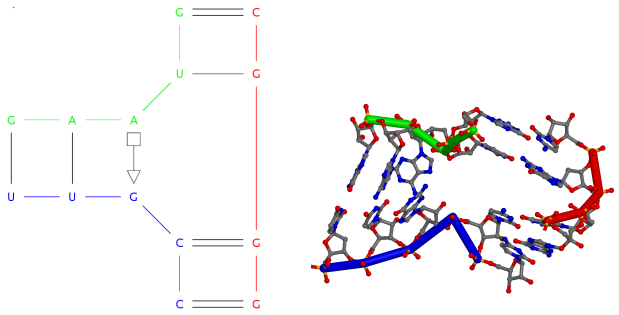
FIG. 8.5 – Le motif structural **2** (jonction de quatre hélices)

Sur (a) on peut voir l'empilement des hélices [19-20 ;117-118] (bleu ;rouge) et [119-120 ;141-142] (rouge ;vert) , et sur (b) l'empilement des hélices [17-18 ;197-198] (bleu ;jaune) et [143-144 ;195-196] (vert ;jaune)



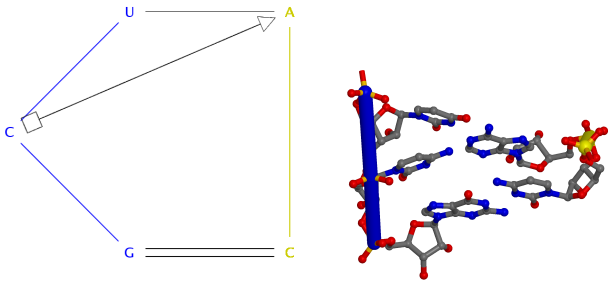
(a) (b) (c)

FIG. 8.6 – Les trois conformations de jonctions à trois [Lescoute06b]



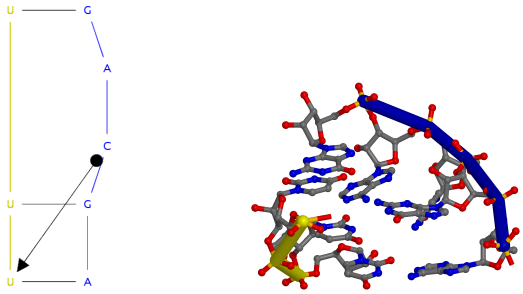
(a) Description (b) Application

FIG. 8.7 – Le motif structural **3** (jonction de trois hélices)



(a) Description (b) Application

FIG. 8.8 – Le motif structural **4** (un nucléotide en bulle)



(a) Description (b) Application

FIG. 8.9 – Le motif structural **5** (deux nucléotides en bulle)

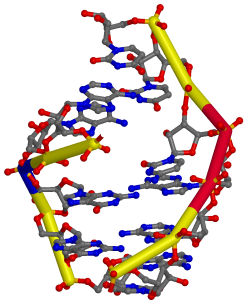
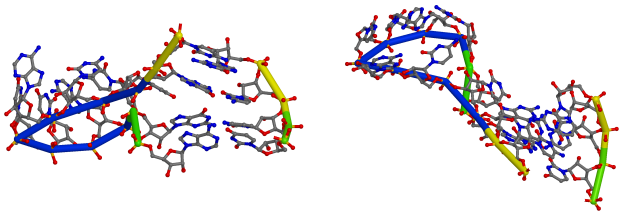


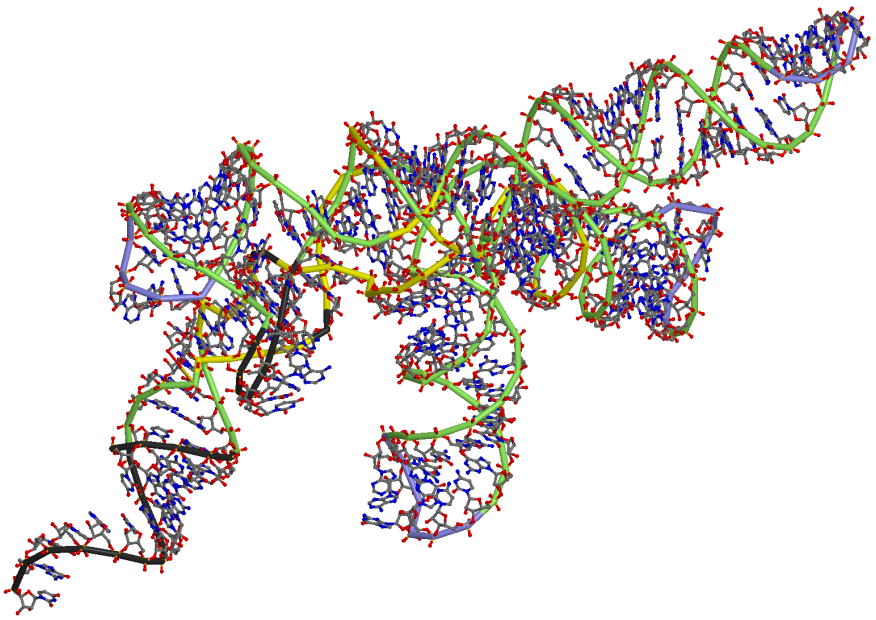
FIG. 8.10 – Les nucléotides 15, 16 et 199



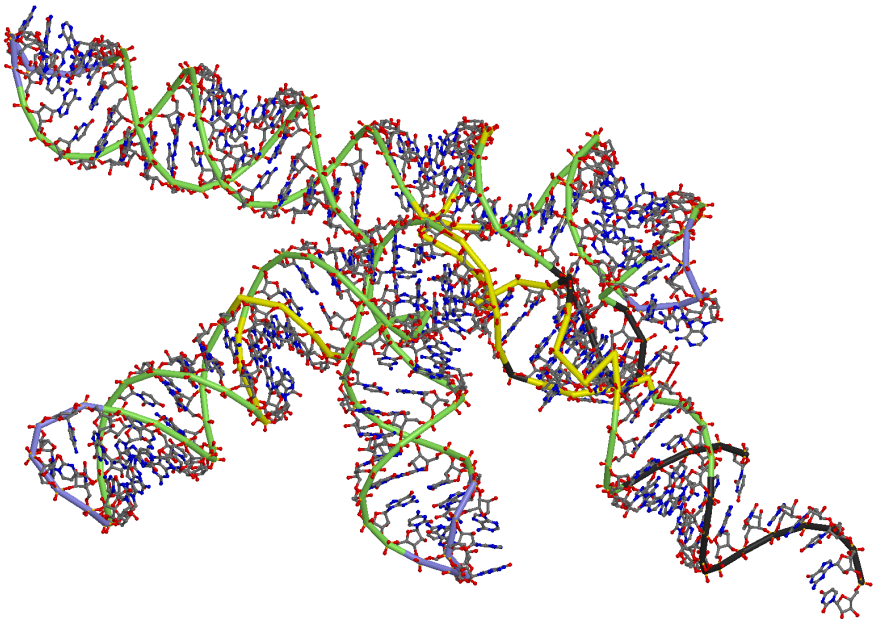
(a) (b)

FIG. 8.11 – Les nucléotides 35 à 41

(a) et (b) montrent deux points de vues de la bulle formée par les nucléotides 35 à 41



(a)



(b)

FIG. 8.12 – Le modèle final du premier état

8.2 La modélisation du second état

8.2.1 L'analyse de la structure secondaire

Les éléments structuraux composant la structure secondaire sont :

- un domaine structural identique entre les deux états de la molécule (α nucléotides 24 à 34 et 42 à 113)
- cinq doubles hélices d'ARN (**H9** à **H13**)
- deux boucles apicales (**E** et **F**)
- trois domaines complexes, correspondants à des motifs structuraux disponibles dans notre banque (**6** à **8**)

8.2.2 La génération des hélices régulières

Sur la figure **8.13**, on peut voir cinq hélices régulières d'ARN. L'hélice **H11** possède une paire de bases non canonique C127-A174.
Une structure secondaire ne décrivant uniquement les appariements des hélices **H9** à **H13** (y compris la paire C127-A174) est fournie à **`Jessa`** et le premier jet de structure est alors généré par **`Nahelix`**.

8.2.3 La modélisation des boucles apicales

La boucle **E** est compatible avec le motif structural GNRA, disponible dans notre banque de motifs, et qui est appliqué à l'aide du **`Motif Repository`**. Le motif GNRA étant destiné à être appliqué sur des fragments de six nucléotides, la boucle **E** comprend les bases U117 et C122 (figure **8.14**).
La boucle **F** comprend sept nucléotides et ne correspond à aucune boucle de notre

banque de motifs. Elle a donc été modélisée manuellement à l'aide de la **`boîte de torsion`** (figure **8.15**).

8.2.4 L'application des motifs structuraux

Trois motifs structuraux disponibles dans notre banque ont pu être identifiés :

- **6**. Deux hélices empilées. Ce motifs composé des nucléotides 114, 126, 127 et 175 permet d'empiler les hélices **H10** et **H11** (figure **8.16**).
- **7**. Une bulle asymétrique de un et deux nucléotides (figure **8.17**).
- **8**. Une bulle de deux fois deux nucléotides (figure **8.18**).

Les repliements et empilements sont réalisés dans l'ordre suivant :

1. Le domaine α [24-34 ;42-113] est replié conformément au modèle du premier état de la molécule, modélisé précédemment
2. **H10** est lié à U113
3. **E** est empilé sur **H10**
4. **6** est empilé sur **H10**
5. **H11** est empilé sur **6**
6. **7** est empilé sur **H11**
7. **H12** est empilé sur**7**
8. **8** est empilé sur **H12**
9. **H13** est empilé sur **8**
10. **F** est empilé sur **H13**

8.2.5 La modélisation manuelle

A ce stade, seules les parties en blanc sur la figure **8.13** ne sont pas modélisées. Les fragments [C16 ;G23] et [U35 ;A37] sont modélisés (figure **8.19**) à l'aide des outils

d'éditions d'**`Assemble`** afin de permettre la jonction entre l'hélice **H9** et le domaine α. Les fragments [U1 ;A11] et [G176 ;U218] ne seront pas modélisés, car on ne dispose d'aucune information structurale à leur égard et qu'ils sont vraisemblablement très flexibles.

Enfin, une étape d'affinement de coordonnées permet de corriger les imperfections du modèle et d'obtenir la structure telle qu'elle peut être visualisée sur la figure **8.20**. La figure **8.21** montre la superposition des deux modèle moléculaires obtenues.

8.2.6 Les figures

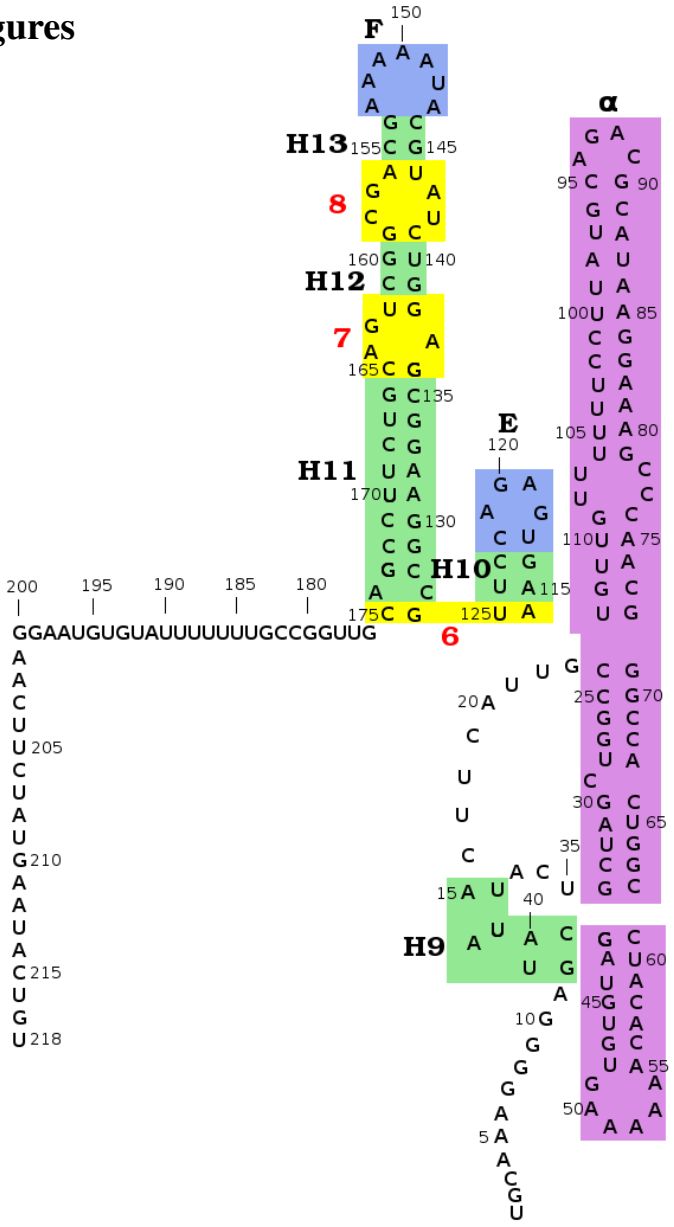
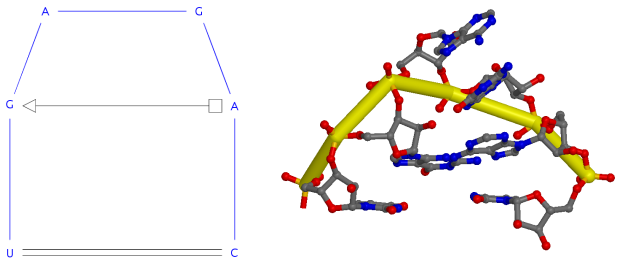


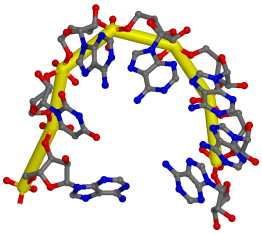
FIG. 8.13 – La structure secondaire du second état à modéliser

En vert : les doubles hélices d'ARN, en bleu : les boucles apicales, en jaune les motifs structuraux et en violet un domaine structural identique entre les deux états de la molécule



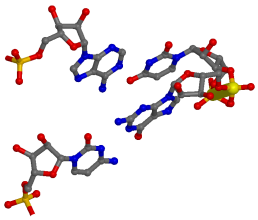
(a) Description (b) Application

FIG. 8.14 – La boucle **E** (GNRA)



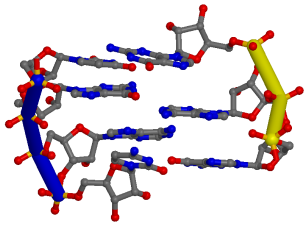
Cette boucle composée majoritairement d'adénines ne correspond pas à un motif structural, elle a été modélisée manuellement.

FIG. 8.15 – La boucle **F**



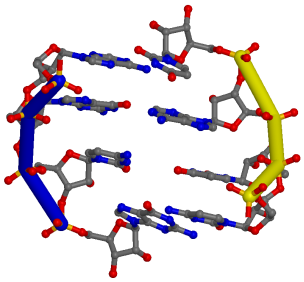
Ce motif est un empilement des paires de bases A114-U125 et G126-C175

FIG. 8.16 – Le motif structural **6**



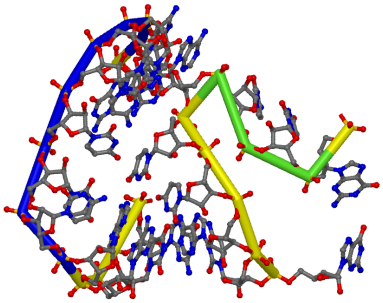
Ce motif est une bulle asymétrique de un et deux nucléotides. Il est composé des brins [G136 ;G138] en jaune et [U162 ;C165] en bleu.

FIG. 8.17 – Le motif structural **7**



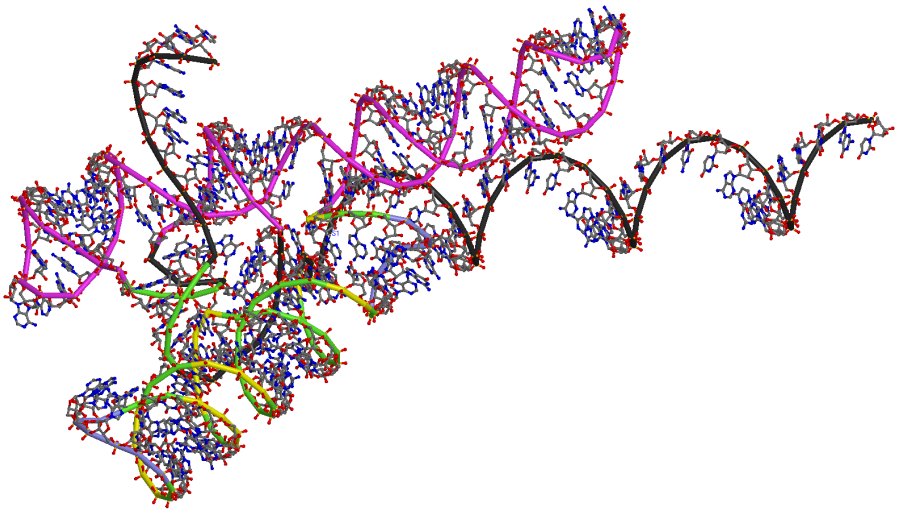
Ce motif est une bulle de deux fois deux nucléotides. Il est composé des brins [C141 ;U144] en jaune et [A156 ;C159] en bleu.

FIG. 8.18 – Le motif structural **8**

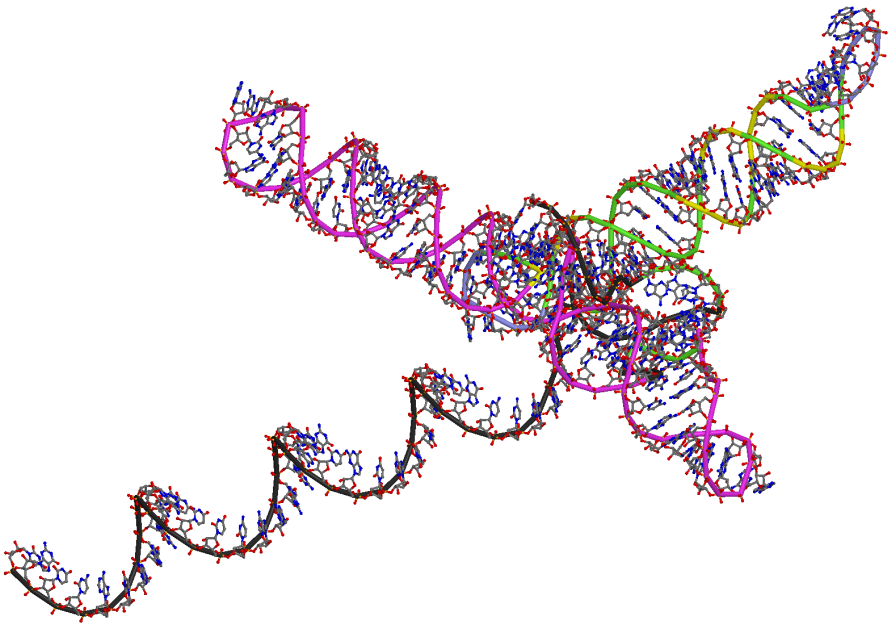


*Le brin [C16 ;G23] en bleu, relie C24 l'hélice **H9** en jaune. Le brin [U35 ;A37] en vert relie G34 et l'hélice **H9***

FIG. 8.19 – Les brins [C16 ;G23] et [U35 ;A37]



(a)



(b)

FIG. 8.20 – Le modèle final du second état

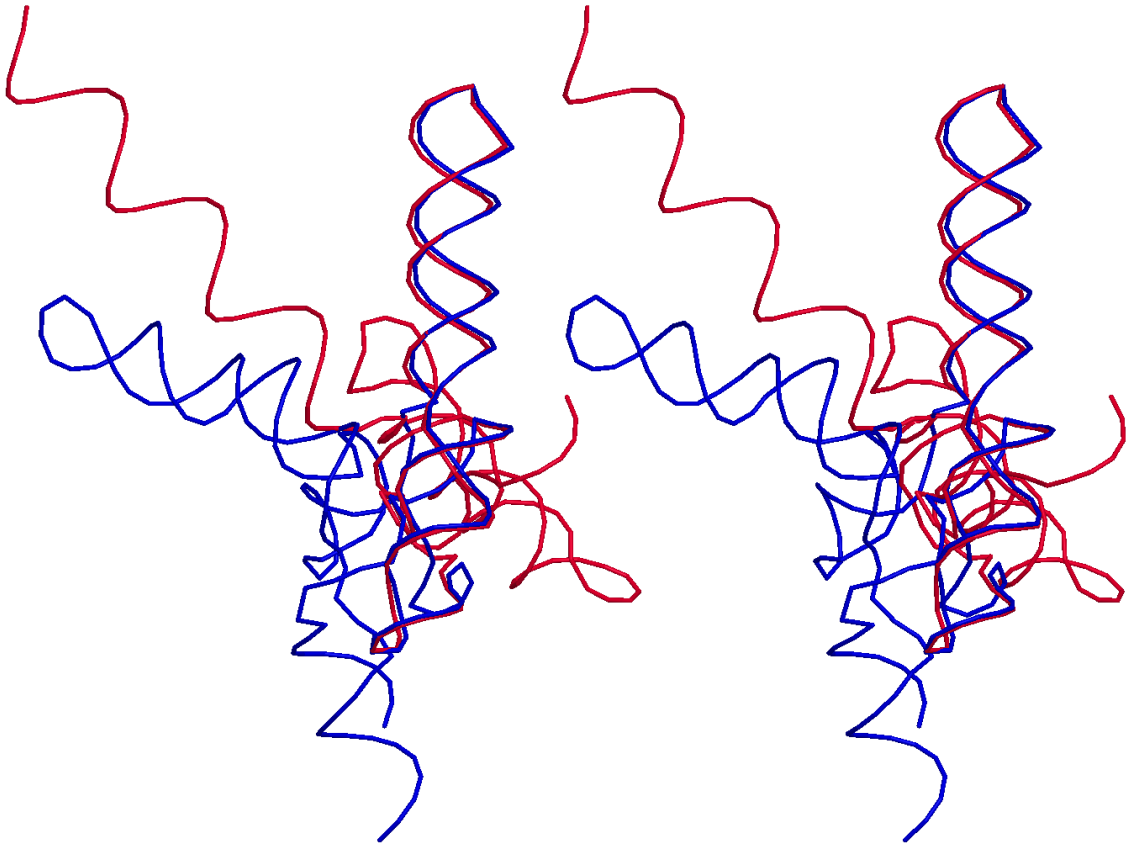


FIG. 8.21 – La superposition des modèles des deux états de la molécule

Les premier et second états sont représentés respectivement en bleu et en rouge

Quatrième partie

Conclusions générales et perspectives

Chapitre 9

Conclusions générales et perspectives

Les travaux de ces dernières années ont permis de montrer que l'ARN est bien plus qu'un vecteur de l'information génétique et qu'il participe à de nombreux mécanismes régulant les grandes fonctions biologiques. De ce fait, l'ARN et les défauts dans sa fonction sont impliqués dans de nombreuses pathologies. La fonction d'un ARN étant déterminée par sa structure, la compréhension des règles structurales de l'ARN est donc essentielle. Cette compréhension ne pourra se faire sans l'analyse de nombreuses structures tridimensionnelles (aujourd'hui encore, peu nombreuses relativement aux molécules séquencées) et le recoupement de l'ensemble des informations disponibles à l'égard des molécules étudiées.

Dans cette optique, cette thèse a présenté le développement d'un environnement bioinformatique dédié à la construction et à la compréhension des architectures d'ARN. Elle décrit l'infrastructure **P.A.R.A.DIS.E**, une évolution du logiciel **S2S** constituant une plateforme d'analyse des annotations d'ARN. Cette plateforme permet à un utilisateur d'analyser, de façon interconnectée, les différentes informations disponibles pour une molécule d'ARN.

La partie **II** a détaillé les trois couches de cette architecture :

- une librairie extensible de concepts relatifs à l'ARN permettant d'intégrer les données manipulées par les différents algorithmes disponibles
- une couche de communication permettant la distribution et l'interconnexion de ces algorithmes
- une couche graphique permettant de visualiser et manipuler les données biologiques et de contrôler les algorithmes

La plateforme **P.A.R.A.DIS.E** propose des outils graphiques adaptés à l'étude de chaque niveau d'analyse de l'ARN, ainsi que des algorithmes permettant de passer d'un niveau à l'autre. Il est donc possible, avec **P.A.R.A.DIS.E**, de confronter l'ensemble des informations dont on dispose vis à vis d'une molécule d'ARN donnée.

L'une des interfaces graphiques proposées par **P.A.R.A.DIS.E**, l'outil **Assemble**, a été décrite en détails dans la partie **III**. **Assemble** permet de construire un modèle de la structure tridimensionnelle d'une molécule d'ARN à partir de sa structure secondaire. Cet outil est adapté pour chaque étape de la construction du modèle, en partant de la structure secondaire, et permet une construction rapide et aisée, grâce à ses nombreux automatismes. La possibilité de générer automatiquement des hélices, d'appliquer des motifs structuraux et d'empiler et assembler ces blocs de construction en quelques clics de souris permet de se soustraire à de nombreuses heures de modélisation manuelle. Il est également possible d'éditer le modèle en respectant les règles stéréochimiques régissant la molécule. L'affichage en arrière plan d'une carte de densité électronique guide l'utilisateur et lui assure à tout moment que le modèle soit bien adéquat avec la réalité biologique de la molécule.

L'intégration d'**Assemble** dans l'infrastructure **P.A.R.A.DIS.E** permet de confronter à tout moment le modèle en cours de création avec les diverses informations disponibles concernant la molécule. Une fois la modélisation achevée, l'outil d'affinement de coordonnées permet d'imposer au modèle le respect de contraintes structurales. Enfin, la panoplie d'aides visuelles, ainsi que l'optimisation des opérations

faites au clavier ou à la souris, rendent l'expérience de modélisation moléculaire à la fois simple, intuitive, efficace et agréable.

Le logiciel **Assemble** a été utilisé dans différents projets de modélisation moléculaire au sein du laboratoire et un exemple, détaillé dans le chapitre **8**, montre qu'il est simple et rapide de modéliser la structure tertiaire d'un ARN avec ce logiciel lorsque la structure secondaire de cette molécule est connue.
L'infrastructure **P.A.R.A.DIS.E** est également adaptée à des projets de modélisation de grandes envergures. En effet, **P.A.R.A.DIS.E** est actuellement utilisé pour modéliser la grande sous unité ribosomale de *S.cerevisiae*. Cette modélisation se fait sur la base d'un alignement structural de cette molécule avec la grande sous unité de *H.marismortui* établi à l'aide de **RNAlign**. Une carte de densité électronique, obtenue par cryo-microscopie électronique et chargée dans **Assemble**, permet de guider la modélisation.

Perspectives

Bien que l'infrastructure **P.A.R.A.DIS.E** et son outil **Assemble** fonctionnent et soient efficaces, comme tout logiciel, ils peuvent encore grandement être améliorés.

Le but à long terme de l'infrastructure **P.A.R.A.DIS.E** est de devenir un LIMS (Laboratory Information Management System) dédié à l'ARN, proposant des analyses toujours plus nombreuses est variées. Pour y parvenir, il est important d'y intégrer l'ensemble des données disponibles pour un ARN : données provenant d'expérimentations humides, contexte génomique, appartenance à des voies métaboliques ou régulatrices, ... La prise en compte de nouveaux types d'informations nécessitera la mise en place :

- de concepts associés à ces informations dans la librairie de concepts de **P.A.R.A.DIS.E**
- d'algorithmes permettant de produire et d'analyser ces informations
- d'interfaces graphiques permettant de visualiser et manipuler ces informations

Sans nécessité l'introduction de nouveaux concepts, l'ajout de nouveaux algorithmes est également utile pour effectuer des analyses impossibles actuellement, et pour proposer des alternatives aux algorithmes présents. En effet, devant la masse d'algorithmes publiés pour une même tâche (voir chapitre **3**), les utilisateurs sont souvent dubitatifs quant au choix du mieux adapté ou du plus efficace pour leurs données. Il peut donc être utile de proposer plusieurs outils pour une tâche telle que l'alignement de séquences et d'aiguiller l'utilisateur en fonction de la nature de ses données (nombre et tailles des séquences, taux de conservation, nombre de séquences annotées d'une structure secondaire ou tertiaire, ...). De plus, pour des algorithmes *a priori* équivalents, **P.A.R.A.DIS.E** pourra servir de «*banc d'essai*» permettant à l'utilisateur de lancer une même analyse en parallèle à l'aide de divers algorithmes candidats et de confronter leurs résultats pour en faire une synthèse et se faire une opinion sur la qualité de chaque algorithme.

Afin d'intégrer de nombreux algorithmes, il est imaginable d'utiliser la technologie fournie par la librairie **JADE** pour donner la possibilité aux agents d'interagir avec des services web. Il est également envisageable d'utiliser pleinement le potentiel d'autonomie des agents afin d'autoriser ces derniers à enchaîner les analyses par eux-mêmes, notamment à l'aide de la librairie **WADE** (**W**orkflow and **A**gent **D**evelopment **E**nvironment) [wade] permettant de mettre en place des enchaînement d'activités réalisées par des agents **JADE**.
Par ailleurs, l'intégration des données d'ARN nécessite de s'orienter vers les nombreuses banques de données existantes qui leurs sont dédiées (voir **3.5**). La possibilité de consulter ces données, directement au sein de **P.A.R.A.DIS.E** représente un atout majeur. Des initiatives sont actuellement en cours au sein de l'équipe pour permettre d'intégrer les informations de banques telles que **RFam**, **RNAdb**, **SCOR** ou **NDB** au sein de **P.A.R.A.DIS.E**. Ces informations seront décrites à l'aide des concepts disponibles dans notre moteur d'annotations et stockées dans une banque de données intégrée à la plateforme **P.A.R.A.DIS.E**.
Enfin, l'ARN ne saurait être pleinement étudié en faisant abstraction du contexte dans lequel il se situe, notamment au sein de la cellule. La génomique et la biologie systémique représentent deux domaines très vastes s'attachant à l'appréhension de ce contexte. Des efforts pour se rapprocher de ces domaines sont à envisager et l'intégration de données protéiques dans l'infrastructure constitue un premier pas important.

Il existe également un grand nombre de fonctionnalités qu'il serait utile d'ajouter à **Assemble**, comme :

- la superposition locale ou globale de deux structures minimisant la RMSD et la possibilité pour l'utilisateur de calculer la RMSD entre deux structures.
- la possibilité de faire appel à des algorithmes et outils utilisés en cristallographie
- l'ajout d'algorithmes permettant de placer de manière semi-automatique des éléments structuraux dans une carte de densité

- l'ajout d'un meilleur algorithme d'affinement de coordonnées permettant de manipuler également des nucléotides modifiés et des protéines
- une extension conséquente de la banque de motifs structuraux
- la possibilité de parcourir des banques telles que **SCOR** afin de trouver des repliements tridimensionnels pour certains motifs structuraux
- la mise en évidence de collisions entre les éléments structuraux
- une option permettant de revenir x étapes en arrière lors de la modélisation
- une amélioration de la qualité du rendu 3D

Alternativement à l'approche de modélisation actuellement proposée par **Assemble**, utilisant les algorithmes **Nahelix** et **Motif Repository**, il peut être intéressant d'intégrer à **P.A.R.A.DIS.E** les algorithmes **FARNA**, **MC-Fold** et **MC-Sym**. Ces algorithmes permettraient d'obtenir des modèles complets d'ARN en une seule étape. La potentielle imperfection de ces modèles pourrait être corrigée avec les outils d'**Assemble** en tenant compte des informations biologiques fournies par **P.A.R.A.DIS.E**.

Bibliographie

[actorfoundary] `http://www-osl.cs.uiuc.edu/foundry`.

[Adams04] PL Adams, MR Stahley, ML Gill, AB Kosek, J Wang & SA Strobel. *Crystal structure of a group I intron splicing intermediate.* RNA, vol. 10, p 1867–87, 2004.

[alignements] `http://pbil.univ-lyon1.fr/alignment.html`.

[Altschul90] SF Altschul, W Gish, W Miller, EW Myers & DJ Lipman. *Basic local alignment search tool.* J Mol Biol, vol. 215, no. 3, p 403–10, 1990.

[Andersen07] E. S. Andersen, A. Lind-Thomsen, B. Knudsen, S. E. Kristensen, J. H. Havgaard, E. Torarinsson, N. Larsen, C. Zwieb, P. Sestoft, J. Kjems & J. Gorodkin. *Semiautomated improvement of RNA alignments*. Rna, vol. 13, no. 11, p 1850–1859, 2007.

[Andrade07] J. Andrade, M. Andersen, A. Sillen, C. Graff & J. Odeberg. *The use of grid computing to drive data-intensive genetic research*. Eur J Hum Genet, vol. 15, no. 6, p 694–702, 2007.

[Andronescu07] M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews & K. P. Murphy. *Efficient parameter estimation for RNA secondary structure prediction*. Bioinformatics, vol. 23, no. 13, p i19–28, 2007.

[Angeletii01] M Angeletii, R Culmone & E Merelli. *An intelligent agent architecture for DNA-microarray data integration*. NETTAB - CORBA and XML : Towards a bioinformatics integrated network environment. Genova.Italy, 2001.

[Anwar06] M. Anwar, T. Nguyen & M. Turcotte. *Identification of consensus RNA secondary structures using suffix arrays*. BMC Bioinformatics, vol. 7, p 244, 2006.

[Armano07] G. Armano, A. Manconi & E. Vargiu. *A multiagent system for retrieving bioinformatics publications from web sources*. IEEE Trans Nanobioscience, vol. 6, no. 2, p 104–109, 2007.

[Ashburner00] M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin & G Sherlock. *Gene ontology : tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, vol. 25, no. 1, p 25–9, 2000.

[Auffinger98] P Auffinger & E Westhof. *Simulations of the molecular dynamics of nucleic acids.* Curr Opin Struct Biol, vol. 8, p 227–36, 1998.

[Auffinger00] P Auffinger & E Westhof. *RNA solvation : a molecular dynamics simulation perspective.* Biopolymers, vol. 56, no. 4, p 266–74, 2000.

[Auffinger07] P Auffinger & Y Hashem. *Nucleic acid solvation : from outside to insight.* Curr Opin Struct Biol, vol. 17, no. 3, p 325–33, 2007.

[Backman08] TW Backman, CM Sullivan, JS Cumbie, ZA Miller, EJ Chapman, N Fahlgren, SA Givan, JC Carrington & KD Kasschau. *Update of ASRP : the Arabidopsis Small RNA Project database.* Nucleic Acids Res, vol. 36, p D982–5, 2008.

[Ban00] N Ban, P Nissen, J Hansen, PB Moore & TA Steitz. *The complete atomic structure of the large ribosomal subunit at 2.4 A resolution.* Science, vol. 289, p 905–20, 2000.

[Beckert08] B Beckert, H Nielsen, C Einvik, SD Johansen, E Westhof & B Masquida. *Molecular modelling of the GIR1 branching ribozyme gives new insight into evolution of structurally related ribozymes.* EMBO J, vol. 27, p 667–78, 2008.

[Bendaña08] YR Bendaña & IH Holmes. *Colorstock, SScolor, Ratón : RNA alignment visualization tools.* Bioinformatics, vol. 24, p 579–80, 2008.

[Berman00] HM Berman, J Westbrook, Z Feng, G Gilliland, TN Bhat, H Weissig, IN Shindyalov & PE Bourne. *The Protein Data Bank.* Nucleic Acids Res, vol. 28, no. 1, p 235–42, 2000.

[Berman02] HM Berman, J Westbrook, Z Feng, L Iype, B Schneider & C Zardecki. *The Nucleic Acid Database.* Acta Crystallogr D Biol Crystallogr, vol. 58, no. Pt 6 No 1, p 889–98, 2002.

[Berman03] HM Berman, J Westbrook, Z Feng, L Iype, B Schneider & C Zardecki. *The nucleic acid database.* Methods Biochem Anal, vol. 44, p 199–216, 2003.

[Bindewald06] E. Bindewald & B. A. Shapiro. *RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers*. Rna, vol. 12, no. 3, p 342–352, 2006.

[biodas] `http://www.biodas.org`.

[biojava] `http://www.biojava.org`.

[biojavainside] `http://www.biojava.org/wiki/BioJava:BioJavaInside`.

[bioperl] `http://www.bioperl.org`.

[biopython] `http://www.biopython.org`.

[bioruby] `http://www.bioruby.org`.

[bmcbioinformatics] `http://www.biomedcentral.com/bmcbioinformatics`.

[Brooks83] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan & M. Karplus. *CHARMM : A Program for Macromolecular Energy, Minimization, and Dynamics Calculations*. J. Comp. Chem., vol. 4, p 187–217, 1983.

[Brown99] JW Brown. *The Ribonuclease P Database.* Nucleic Acids Res, vol. 27, p 314, 1999.

[Brunger07] AT Brunger. *Version 1.2 of the Crystallography and NMR system.* Nat Protoc, vol. 2, no. 11, p 2728–33, 2007.

[Bryson00a] K Bryson, M Luck, M Joy & D Jones. *Applying agents to bioinformatics in GeneWeaver*. Cooperative information agents IV. Lecture notes in artif. intell., vol. 1860, p 60–71, 2000.

[Bryson00b] K Bryson, M Luck, M Joy, D Jones, P Nicholas & P Bessieres. *From geneweaver to agmial*. Network tools and applications in biology—agents in bioinformatics, 2000.

[Burks05] J Burks, C Zwieb, F Müller, I Wower & J Wower. *Comparative 3-D modeling of tmRNA*. BMC Mol Biol, vol. 6, p 14, 2005.

[Cai03] L Cai, RL Malmberg & Y Wu. *Stochastic modeling of RNA pseudoknotted structures : a grammatical approach.* Bioinformatics, vol. 19 Suppl 1, p i66–73, 2003.

[Case05] DA Case, TE 3rd Cheatham, T Darden, H Gohlke, R Luo, KM Jr Merz, A Onufriev, C Simmerling, B Wang & RJ Woods. *The Amber biomolecular simulation programs.* J Comput Chem, vol. 26, no. 16, p 1668–88, 2005.

[CCP494] CCP4. *The CCP4 suite : programs for protein crystallography.* Acta Crystallogr D Biol Crystallogr, vol. 50, no. Pt 5, p 760–3, 1994.

[Chan05] CY Chan, CE Lawrence & Y Ding. *Structure clustering features on the Sfold Web server.* Bioinformatics, vol. 21, no. 20, p 3926–8, 2005.

[Chang06] T. H. Chang, H. D. Huang, T. N. Chuang, D. M. Shien & J. T. Horng. *RNAMST : efficient and flexible approach for identifying RNA structural homologs*. Nucleic Acids Res, vol. 34, no. Web Server issue, p W423–8, 2006.

[Chetouani97] F Chetouani, P Monestié, P Thébault, C Gaspin & B Michot. *ESSA : an integrated and interactive computer tool for analysing RNA secondary structure.* Nucleic Acids Res, vol. 25, p 3514–22, 1997.

[Chiromatzo07] AO Chiromatzo, TY Oliveira, G Pereira, AY Costa, CA Montesco, DE Gras, F Yosetake, JB Vilar, M Cervato, PR Prado, RG Cardenas, R Cerri, RL Borges, RN Lemos, SM Alvarenga, VR Perallis, DG Pinheiro, IT Silva, RM Brandão, MA Cunha, S Giuliatti & WA Jr Silva. *miRNApath : a database of miRNAs, target genes and metabolic pathways.* Genet Mol Res, vol. 6, no. 4, p 859–65, 2007.

[Clemons01] WM Jr Clemons, DE Brodersen, JP McCutcheon, JL May, AP Carter, RJ Morgan-Warren, BT Wimberly & V Ramakrishnan. *Crystal structure of the 30 S ribosomal subunit from Thermus thermophilus : purification, crystallization and structure determination.* J Mol Biol, vol. 310, no. 4, p 827–43, 2001.

[Cojocaru05] V Cojocaru, S Nottrott, R Klement & TM Jovin. *The snRNP 15.5K protein folds its cognate K-turn RNA : a combined theoretical and biochemical study.* RNA, vol. 11, p 197–209, 2005.

[Colbourn07] C. J. Colbourn & S. Kumar. *Lower bounds on multiple sequence alignment using exact 3-way alignment*. BMC Bioinformatics, vol. 8, p 140, 2007.

[Costa95] M Costa & F Michel. *Frequent use of the same tertiary motif by self-folding RNAs.* EMBO J, vol. 14, p 1276–85, 1995.

[Costa97] M Costa & F Michel. *Rules for RNA recognition of GNRA tetraloops deduced by in vitro selection : comparison with in vivo evolution.* EMBO J, vol. 16, p 3289–302, 1997.

[Costa00] M Costa, F Michel & E Westhof. *A three-dimensional perspective on exon binding by a group II self-splicing intron.* EMBO J, vol. 19, p 5007–18, 2000.

[cougaar] `http://www.cougaar.org`.

[Coventry04] A. Coventry, D. J. Kleitman & B. Berger. *MSARI : multiple sequence alignments for statistical detection of RNA secondary structure*. Proc Natl Acad Sci U S A, vol. 101, no. 33, p 12102–12107, 2004.

[Covitz03] PA Covitz, F Hartel, C Schaefer, S De Coronado, G Fragoso, H Sahni, S Gustafson & KH Buetow. *caCORE : a common infrastructure for cancer informatics.* Bioinformatics, vol. 19, no. 18, p 2404–12, 2003.

[Curcin05] V Curcin, M Ghanem & Y Guo. *Web services in the life sciences.* Drug Discov Today, vol. 10, no. 12, p 865–71, 2005.

[Dalli06] D. Dalli, A. Wilm, I. Mainz & G. Steger. *STRAL : progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time*. Bioinformatics, vol. 22, no. 13, p 1593–1599, 2006.

[Das07] R. Das & D. Baker. *Automated de novo prediction of native-like RNA tertiary structures*. Proc Natl Acad Sci U S A, 2007.

[De Rijk97] P De Rijk & R De Wachter. *RnaViz, a program for the visualisation of RNA secondary structure.* Nucleic Acids Res, vol. 25, no. 22, p 4679–84, 1997.

[De Rijk03] P De Rijk, J Wuyts & R De Wachter. *RnaViz 2 : an improved representation of RNA secondary structure.* Bioinformatics, vol. 19, no. 2, p 299–300, 2003.

[Decker02] K Decker, S Khan & C Schmidt. *BioMAS : A multi-agent system for genomic annotation*. Int. J. Cooperative Inform.Systems, vol. 11, p 265–292, 2002.

[Delcher02] AL Delcher, A Phillippy, J Carlton & SL Salzberg. *Fast algorithms for large-scale genome alignment and comparison.* Nucleic Acids Res, vol. 30, no. 11, p 2478–83, 2002.

[Dezulian06] T. Dezulian, M. Remmert, J. F. Palatnik, D. Weigel & D. H. Huson. *Identification of plant microRNA homologs*. Bioinformatics, vol. 22, no. 3, p 359–360, 2006.

[di Bernardo03] D. di Bernardo, T. Down & T. Hubbard. *ddbRNA : detection of conserved secondary structures in multiple alignments*. Bioinformatics, vol. 19, no. 13, p 1606–1611, 2003.

[Ding05] Y. Ding, C. Y. Chan & C. E. Lawrence. *RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble*. Rna, vol. 11, no. 8, p 1157–1166, 2005.

[Dirks03] RM Dirks & NA Pierce. *A partition function algorithm for nucleic acid secondary structure including pseudoknots.* J Comput Chem, vol. 24, no. 13, p 1664–77, 2003.

[Do06] C. B. Do, D. A. Woods & S. Batzoglou. *CONTRAfold : RNA secondary structure prediction without physics-based models*. Bioinformatics, vol. 22, no. 14, p e90–8, 2006.

[Doudna02] JA Doudna & TR Cech. *The chemical repertoire of natural ribozymes.* Nature, vol. 418, p 222–8, 2002.

[Dowell06] R. D. Dowell & S. R. Eddy. *Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints*. BMC Bioinformatics, vol. 7, p 400, 2006.

[Dror05] O Dror, R Nussinov & H Wolfson. *ARTS : alignment of RNA tertiary structures.* Bioinformatics, vol. 21 Suppl 2, p ii47–53, 2005.

[Dror06] O. Dror, R. Nussinov & H. J. Wolfson. *The ARTS web server for aligning RNA tertiary structures*. Nucleic Acids Res, vol. 34, no. Web Server issue, p W412–5, 2006.

[Duarte03] CM Duarte, LM Wadley & AM Pyle. *RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space.* Nucleic Acids Res, vol. 31, p 4755–61, 2003.

[Dykxhoorn08] DM Dykxhoorn, D Chowdhury & J Lieberman. *RNA interference and cancer : endogenous pathways and therapeutic approaches.* Adv Exp Med Biol, vol. 615, p 299–329, 2008.

[Eckart03] JD Eckart & BW Sobral. *A life scientist's gateway to distributed data management and computing : the PathPort/ToolBus framework.* OMICS, vol. 7, no. 1, p 79–88, 2003.

[Eddy96] SR Eddy. *Hidden Markov models.* Curr Opin Struct Biol, vol. 6, no. 3, p 361–5, 1996.

[Eddy98] SR Eddy. *Profile hidden Markov models.* Bioinformatics, vol. 14, no. 9, p 755–63, 1998.

[Eddy04] SR Eddy. *What is a hidden Markov model ?* Nat Biotechnol, vol. 22, no. 10, p 1315–6, 2004.

[Eddy06] S. R. Eddy. *Computational analysis of RNAs*. Cold Spring Harb Symp Quant Biol, vol. 71, p 117–128, 2006.

[Engelen07] S Engelen & F Tahi. *Predicting RNA secondary structure by the comparative approach : how to select the homologous sequences.* BMC Bioinformatics, vol. 8, no. 1, p 464, 2007.

[esoap] `http://www.ncbi.nlm.nih.gov/entrez/query/static/esoap_help.html`.

[Felden07] B Felden. *RNA structure : experimental analysis.* Curr Opin Microbiol, vol. 10, no. 3, p 286–91, 2007.

[Fiers08] MW Fiers, A van der Burgt, E Datema, JC de Groot & RC van Ham. *High-throughput bioinformatics with the Cyrille2 pipeline system.* BMC Bioinformatics, vol. 9, no. 1, p 96, 2008.

[fipa] `http://www.fipa.org`.

[Flicek08] P Flicek, BL Aken, K Beal, B Ballester, M Caccamo, Y Chen, L Clarke, G Coates, F Cunningham, T Cutts, T Down, SC Dyer, T Eyre, S Fitzgerald, J Fernandez-Banet, S Gräf, S Haider, M Hammond, R Holland, KL Howe, K Howe, N Johnson, A Jenkinson, A Kähäri, D Keefe, F Kokocinski, E Kulesha, D Lawson, I Longden, K Megy, P Meidl, B Overduin, A Parker, B Pritchard, A Prlic, S Rice, D Rios, M Schuster, I Sealy, G Slater, D Smedley, G Spudich, S Trevanion, AJ Vilella, J Vogel, S White, M Wood, E Birney, T Cox, V Curwen, R Durbin, XM Fernandez-Suarez, J Herrero, TJ Hubbard, A Kasprzyk, G Proctor, J Smith, A Ureta-Vidal & S Searle. *Ensembl 2008.* Nucleic Acids Res, vol. 36, no. Database issue, p D707–14, 2008.

[Foster05] I Foster. *Service-oriented science.* Science, vol. 308, no. 5723, p 814–7, 2005.

[Freyhult05] E. Freyhult, V. Moulton & P. Gardner. *Predicting RNA structure using mutual information*. Appl Bioinformatics, vol. 4, no. 1, p 53–59, 2005.

[Garcia Castro05] A Garcia Castro, S Thoraval, LJ Garcia & MA Ragan. *Workflows in bioinformatics : meta-analysis and prototype implementation of a workflow generator.* BMC Bioinformatics, vol. 6, p 87, 2005.

[Gardner04] P. P. Gardner & R. Giegerich. *A comprehensive comparison of comparative RNA structure prediction approaches*. BMC Bioinformatics, vol. 5, p 140, 2004.

[Gardner05] P. P. Gardner, A. Wilm & S. Washietl. *A benchmark of multiple sequence alignment programs upon structural RNAs*. Nucleic Acids Res, vol. 33, no. 8, p 2433–2439, 2005.

[Gewehr07] J. E. Gewehr, M. Szugat & R. Zimmer. *BioWeka–extending the Weka framework for bioinformatics*. Bioinformatics, vol. 23, no. 5, p 651–653, 2007.

[Gondro07] C Gondro & BP Kinghorn. *A simple genetic algorithm for multiple sequence alignment.* Genet Mol Res, vol. 6, no. 4, p 964–82, 2007.

[Goode07] M. G. Goode & A. G. Rodrigo. *SQUINT : a multiple alignment program and editor*. Bioinformatics, vol. 23, no. 12, p 1553–1555, 2007.

[Goody04] TA Goody, SE Melcher, DG Norman & DM Lilley. *The kink-turn motif in RNA is dimorphic, and metal ion-dependent.* RNA, vol. 10, no. 2, p 254–64, 2004.

[Griffiths-Jones03] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna & S. R. Eddy. *Rfam : an RNA family database*. Nucleic Acids Res, vol. 31, no. 1, p 439–441, 2003.

[Griffiths-Jones05a] S Griffiths-Jones. *RALEE–RNA ALignment editor in Emacs.* Bioinformatics, vol. 21, no. 2, p 257–9, 2005.

[Griffiths-Jones05b] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S. R. Eddy & A. Bateman. *Rfam : annotating non-coding RNAs in complete genomes*. Nucleic Acids Res, vol. 33, no. Database issue, p D121–4, 2005.

[Grillo03] G. Grillo, F. Licciulli, S. Liuni, E. Sbisa & G. Pesole. *PatSearch : A program for the detection of patterns and structural motifs in nucleotide sequences*. Nucleic Acids Res, vol. 31, no. 13, p 3608–3612, 2003.

[Gruber93] T. R. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, vol. 5, p 199–200, 1993.

[Gruber08] AR Gruber, SH Bernhart, IL Hofacker & S Washietl. *Strategies for measuring evolutionary conservation of RNA secondary structures.* BMC Bioinformatics, vol. 9, no. 1, p 122, 2008.

[Gudbjartsson00] DF Gudbjartsson, K Jonasson, ML Frigge & A Kong. *Allegro, a new computer program for multipoint linkage analysis.* Nat Genet, vol. 25, no. 1, p 12–3, 2000.

[Gudbjartsson05] DF Gudbjartsson, T Thorvaldsson, A Kong, G Gunnarsson & A Ingolfsdottir. *Allegro version 2.* Nat Genet, vol. 37, no. 10, p 1015–6, 2005.

[Gultyaev95] AP Gultyaev, FH van Batenburg & CW Pleij. *The computer simulation of RNA folding pathways using a genetic algorithm.* J Mol Biol, vol. 250, p 37–51, 1995.

[Gustafson05] A. M. Gustafson, E. Allen, S. Givan, D. Smith, J. C. Carrington & K. D. Kasschau. *ASRP : the Arabidopsis Small RNA Project Database*. Nucleic Acids Res, vol. 33, no. Database issue, p D637–40, 2005.

[Hamada06] M. Hamada, K. Tsuda, T. Kudo, T. Kin & K. Asai. *Mining frequent stem patterns from unaligned RNA sequences*. Bioinformatics, vol. 22, no. 20, p 2480–2487, 2006.

[Han03] K Han & Y Byun. *PSEUDOVIEWER2 : Visualization of RNA pseudoknots of any type.* Nucleic Acids Res, vol. 31, no. 13, p 3432–40, 2003.

[Hansen07] TM Hansen, SN Reihani, LB Oddershede & MA Sørensen. *Correlation between mechanical strength of messenger RNA pseudoknots and ribosomal frameshifting.* Proc Natl Acad Sci U S A, vol. 104, p 5830–5, 2007.

[Harmanci07] A. O. Harmanci, G. Sharma & D. H. Mathews. *Efficient pairwise RNA structure prediction using probabilistic alignment constraints in Dynalign*. BMC Bioinformatics, vol. 8, p 130, 2007.

[Harms01] J Harms, F Schluenzen, R Zarivach, A Bashan, S Gat, I Agmon, H Bartels, F Franceschi & A Yonath. *High resolution structure of the large ribosomal subunit from a mesophilic eubacterium.* Cell, vol. 107, p 679–88, 2001.

[Harrison03] AM Harrison, DR South, P Willett & PJ Artymiuk. *Representation, searching and discovery of patterns of bases in complex RNA structures.* J Comput Aided Mol Des, vol. 17, p 537–49, 2003.

[Havgaard05] J. H. Havgaard, R. B. Lyngso & J. Gorodkin. *The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search*. Nucleic Acids Res, vol. 33, no. Web Server issue, p W650–3, 2005.

[Havgaard07] JH Havgaard, E Torarinsson & J Gorodkin. *Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix.* PLoS Comput Biol, vol. 3, p 1896–908, 2007.

[He08] S He, C Liu, G Skogerbø, H Zhao, J Wang, T Liu, B Bai, Y Zhao & R Chen. *NONCODE v2.0 : decoding the non-coding.* Nucleic Acids Res, vol. 36, p D170–2, 2008.

[Hershkovitz03] E Hershkovitz, E Tannenbaum, SB Howerton, A Sheth, A Tannenbaum & LD Williams. *Automated identification of RNA conformational motifs : theory and application to the HM LSU 23S rRNA*. Nucleic Acids Res, vol. 31, p 6249–57, 2003.

[Hershkovitz06] E Hershkovitz, G Sapiro, A Tannenbaum & LD Williams. *Statistical analysis of RNA backbone.* IEEE/ACM Trans Comput Biol Bioinform, vol. 3, no. 1, p 33–46, 2006.

[Hertel06] J Hertel & PF Stadler. *Hairpins in a Haystack : recognizing microRNA precursors in comparative genomics data.* Bioinformatics, vol. 22, no. Precursors, p e197–202, 2006.

[Hofacker94] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M. Tacker & P. Schuste. *Fast Folding and Comparison of RNA Secondary Structures*. Monatshefte f. Chemie, vol. 125, p 167–188, 1994.

[Hofacker03] I. L. Hofacker. *Vienna RNA secondary structure server*. Nucleic Acids Res, vol. 31, no. 13, p 3429–3431, 2003.

[Hofacker07] IL Hofacker. *RNA consensus structure prediction with RNAalifold.* Methods Mol Biol, vol. 395, p 527–44, 2007.

[Holmes05] I Holmes. *Accelerated probabilistic inference of RNA structure evolution.* BMC Bioinformatics, vol. 6, p 73, 2005.

[Hoogsteen63] Karst Hoogsteen. *The crystal and molecular structure of a hydrogen-bonded complex between 1-methylthymine and 9-methyladenine*. Acta Crystallogr, vol. 16, p 907–916, 1963.

[Horesh07] Y. Horesh, T. Doniger, S. Michaeli & R. Unger. *RNAspa : a shortest path approach for comparative prediction of the secondary structure of ncRNA molecules*. BMC Bioinformatics, vol. 8, no. 1, p 366, 2007.

[Huang96] C.C. Huang, G.S. Couch, E.F. Pettersen & T.E Ferrin. *Chimera : An Extensible Molecular Modeling Application Constructed Using Standard Components*. Pacific Symposium on Biocomputing, vol. 1, p 724, 1996.

[Huang05] H. C. Huang, U. Nagaswamy & G. E. Fox. *The application of cluster analysis in the intercomparison of loop structures in RNA*. Rna, vol. 11, no. 4, p 412–423, 2005.

[Huang08] DD Huang. *The potential of RNA interference-based therapies for viral infections.* Curr HIV/AIDS Rep, vol. 5, p 33–9, 2008.

[Humphrey96] W Humphrey, A Dalke & K Schulten. *VMD : visual molecular dynamics.* J Mol Graph, vol. 14, no. 1, p 33–8, 27–8, 1996.

[Höchsmann03] M Höchsmann, T Töller, R Giegerich & S Kurtz. *Local similarity in RNA secondary structures.* Proc IEEE Comput Soc Bioinform Conf, vol. 2, p 159–68, 2003.

[jade] `http://jade.tilab.com`.

[Jaeger94] L Jaeger, F Michel & E Westhof. *Involvement of a GNRA tetraloop in long-range RNA tertiary interactions.* J Mol Biol, vol. 236, p 1271–6, 1994.

[Jansson06] J. Jansson, N. T. Hieu & W. K. Sung. *Local gapped subforest alignment and its application in finding RNA structural motifs*. J Comput Biol, vol. 13, no. 3, p 702–718, 2006.

[jason] `http://jason.sourceforge.net`.

[java] `http://java.sun.com`.

[java3d] `http://java3d.dev.java.net`.

[Jeon05] YS Jeon, H Chung, S Park, I Hur, JH Lee & J Chun. *jPHYDIT : a JAVA-based integrated environment for molecular phylogeny of ribosomal RNA sequences.* Bioinformatics, vol. 21, p 3171–3, 2005.

[jogl] `https://jogl.dev.java.net`.

[Johnson08] M Johnson, I Zaretskaya, Y Raytselis, Y Merezhuk, S McGinnis & TL Madden. *NCBI BLAST : a better web interface.* Nucleic Acids Res, 2008.

[Jones85] TA Jones. *Diffraction methods for biological macromolecules. Interactive computer graphics : FRODO.* Methods Enzymol, vol. 115, p 157–71, 1985.

[Jossinet05] F. Jossinet & E. Westhof. *Sequence to Structure (S2S) : display, manipulate and interconnect RNA data from sequence to structure*. Bioinformatics, vol. 21, no. 15, p 3320–3321, 2005.

[Jossinet07] F. Jossinet, T. E. Ludwig & E. Westhof. *RNA structure : bioinformatic analysis*. Curr Opin Microbiol, vol. 10, no. 3, p 279–285, 2007.

[Jucker95] FM Jucker & A Pardi. *GNRA tetraloops make a U-turn.* RNA, vol. 1, no. 2, p 219–22, 1995.

[Karasavvas02] K Karasavvas, A Burger & RA Baldock. *A multi-agent bioinformatics integration system with adjustable autonomy*. PRICAI,Lecture Notes in Computer Science, vol. 2417, p 492–501, 2002.

[Karasavvas04] KA Karasavvas, R Baldock & A Burger. *Bioinformatics integration and agent technology.* J Biomed Inform, vol. 37, no. 3, p 205–19, 2004.

[Kawas06] E Kawas, M Senger & MD Wilkinson. *BioMoby extensions to the Taverna workflow management and enactment software.* BMC Bioinformatics, vol. 7, p 523, 2006.

[Keele05] JW Keele & JE Wray. *Software agents in molecular computational biology.* Brief Bioinform, vol. 6, no. 4, p 370–9, 2005.

[keggapi] `http://www.genome.jp/kegg/soap`.

[Khan03] S Khan, R Makkena, F McGeary, K Decker, W Gillis & C Schmidt. *A multi-agent system for the quantitative simulation of biological networks*. Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, p 385–392, 2003.

[Kin07] T Kin, K Yamada, G Terai, H Okida, Y Yoshinari, Y Ono, A Kojima, Y Kimura, T Komori & K Asai. *fRNAdb : a platform for mining/annotating functional RNA*

candidates from non-coding RNA sequences. Nucleic Acids Res, vol. 35, p D145–8, 2007.

[Kiryu07] H. Kiryu, Y. Tabei, T. Kin & K. Asai. *Murlet : a practical multiple alignment tool for structural RNA sequences*. Bioinformatics, vol. 23, no. 13, p 1588–1598, 2007.

[Klein01] DJ Klein, TM Schmeing, PB Moore & TA Steitz. *The kink-turn : a new RNA secondary structure motif.* EMBO J, vol. 20, p 4214–21, 2001.

[Klein03] R. J. Klein & S. R. Eddy. *RSEARCH : finding homologs of single structured RNA sequences*. BMC Bioinformatics, vol. 4, p 44, 2003.

[Klosterman02] P. S. Klosterman, M. Tamura, S. R. Holbrook & S. E. Brenner. *SCOR : a Structural Classification of RNA database*. Nucleic Acids Res, vol. 30, no. 1, p 392–394, 2002.

[Knight04] R. Knight, A. Birmingham & M. Yarus. *BayesFold : rational 2 degrees folds that combine thermodynamic, covariation, and chemical data for aligned RNA sequences*. Rna, vol. 10, no. 9, p 1323–1336, 2004.

[Knudsen03] B. Knudsen & J. Hein. *Pfold : RNA secondary structure prediction using stochastic context-free grammars*. Nucleic Acids Res, vol. 31, no. 13, p 3423–3428, 2003.

[Kochiwa06] H. Kochiwa, A. Kanai & T. Masaru. *[Bioinformatics analyses of non-coding RNA]*. Tanpakushitsu Kakusan Koso, vol. 51, no. 16 Suppl, p 2420–2424, 2006.

[Komatsoulis08] GA Komatsoulis, DB Warzel, FW Hartel, K Shanbhag, R Chilukuri, G Fragoso, S Coronado, DM Reeves, JB Hadfield, C Ludet & PA Covitz. *caCORE version 3 : Implementation of a model driven, service-oriented architecture for semantic interoperability.* J Biomed Inform, vol. 41, no. 1, p 106–23, 2008.

[Konnert80] J. H. Konnert & W. A. Hendrickson. *A restrained-parameter. thermal-factor refinement procedur*. Acta Crystallogr A, vol. 36, p 344–350, 1980.

[Krasilnikov04] AS Krasilnikov, Y Xiao, T Pan & A Mondragón. *Basis for structural diversity in homologous RNAs.* Science, vol. 306, p 104–7, 2004.

[Krol90] A Krol, E Westhof, M Bach, R Lührmann, JP Ebel & P Carbon. *Solution structure of human U1 snRNA. Derivation of a possible three-dimensional model.* Nucleic Acids Res, vol. 18, p 3803–11, 1990.

[Kruspe07] M. Kruspe & P. F. Stadler. *Progressive multiple sequence alignments from triplets*. BMC Bioinformatics, vol. 8, p 254, 2007.

[Kuhn02] JF Kuhn, EJ Tran & ES Maxwell. *Archaeal ribosomal protein L7 is a functional homolog of the eukaryotic 15.5kD/Snu13p snoRNP core protein.* Nucleic Acids Res, vol. 30, p 931–41, 2002.

[Kumar08] A Kumar & KT Jeang. Insights into cellular micrornas and human immunodeficiency virus type 1 (hiv-1)., volume 216. -1, 2008.

[Labarga07] A. Labarga, F. Valentin, M. Anderson & R. Lopez. *Web services at the European bioinformatics institute*. Nucleic Acids Res, vol. 35, no. Web Server issue, p W6–11, 2007.

[Lambert04] A Lambert, JF Fontaine, M Legendre, F Leclerc, E Permal, F Major, H Putzer, O Delfour, B Michot & D Gautheret. *The ERPIN server : an interface to profile-based RNA motif identification.* Nucleic Acids Res, vol. 32, no. Web Server issue, p W160–5, 2004.

[Lanzén08] A Lanzén & T Oinn. *The Taverna Interaction Service : enabling manual interaction in workflows.* Bioinformatics, vol. 24, no. 8, p 1118–20, 2008.

[Le04] S. Y. Le, Jr. Maizel J. V. & K. Zhang. *An algorithm for detecting homologues of known structured RNAs in genomes*. Proc IEEE Comput Syst Bioinform Conf, p 300–310, 2004.

[Lehnert96] V Lehnert, L Jaeger, F Michel & E Westhof. *New loop-loop tertiary interactions in self-splicing introns of subgroup IC and ID : a complete 3D model of the Tetrahymena thermophila ribozyme.* Chem Biol, vol. 3, p 993–1009, 1996.

[Lemieux02] S Lemieux & F Major. *RNA canonical and non-canonical base pairing types : a recognition method and complete repertoire.* Nucleic Acids Res, vol. 30, p 4250–63, 2002.

[Lemieux06] S. Lemieux & F. Major. *Automated extraction and classification of RNA tertiary structure cyclic motifs*. Nucleic Acids Res, vol. 34, no. 8, p 2340–2346, 2006.

[Leontis01] NB Leontis & E Westhof. *Geometric nomenclature and classification of RNA base pairs.* RNA, vol. 7, p 499–512, 2001.

[Leontis02a] NB Leontis, J Stombaugh & E Westhof. *Motif prediction in ribosomal RNAs Lessons and prospects for automated motif prediction in homologous RNA molecules.* Biochimie, vol. 84, p 961–73, 2002.

[Leontis02b] NB Leontis, J Stombaugh & E Westhof. *The non-Watson-Crick base pairs and their associated isostericity matrices.* Nucleic Acids Res, vol. 30, no. 16, p 3497–531, 2002.

[Leontis03] NB Leontis & E Westhof. *Analysis of RNA motifs.* Curr Opin Struct Biol, vol. 13, p 300–8, 2003.

[Leontis06a] NB Leontis, RB Altman, HM Berman, SE Brenner, JW Brown, DR Engelke, SC Harvey, SR Holbrook, F Jossinet, SE Lewis, F Major, DH Mathews, JS Richardson, JR Williamson & E Westhof. *The RNA Ontology Consortium : an open invitation to the RNA community.* RNA, vol. 12, no. 4, p 533–41, 2006.

[Leontis06b] NB Leontis, A Lescoute & E Westhof. *The building blocks and motifs of RNA architecture.* Curr Opin Struct Biol, vol. 16, no. 3, p 279–87, 2006.

[Lescoute05] A Lescoute, NB Leontis, C Massire & E Westhof. *Recurrent structural RNA motifs, Isostericity Matrices and sequence alignments.* Nucleic Acids Res, vol. 33, p 2395–409, 2005.

[Lescoute06a] A Lescoute & E Westhof. *The interaction networks of structured RNAs.* Nucleic Acids Res, vol. 34, p 6587–604, 2006.

[Lescoute06b] A Lescoute & E Westhof. *Topology of three-way junctions in folded RNAs.* RNA, vol. 12, no. 1, p 83–93, 2006.

[Lindgreen07] S. Lindgreen, P. P. Gardner & A. Krogh. *MASTR : Multiple alignment and structure prediction of non-coding RNAs using simulated annealing*. Bioinformatics, 2007.

[Liu05a] C Liu, B Bai, G Skogerbø, L Cai, W Deng, Y Zhang, D Bu, Y Zhao & R Chen. *NONCODE : an integrated knowledge database of non-coding RNAs.* Nucleic Acids Res, vol. 33, p D112–5, 2005.

[Liu05b] J. Liu, J. T. Wang, J. Hu & B. Tian. *A method for aligning RNA secondary structures and its application to RNA motif detection*. BMC Bioinformatics, vol. 6, p 89, 2005.

[Liu06] H. Liu, D. Xu, J. Shao & Y. Wang. *An RNA folding algorithm including pseudoknots based on dynamic weighted matching*. Comput Biol Chem, vol. 30, no. 1, p 72–76, 2006.

[Liu08] Q Liu, V Olman, H Liu, X Ye, S Qiu & Y Xu. *RNACluster : An integrated tool for RNA secondary structure comparison and clustering.* J Comput Chem, 2008.

[Lorenz08] WA Lorenz, Y Ponty & P Clote. *Asymptotics of RNA Shapes.* J Comput Biol, 2008.

[Louise-May96] S Louise-May, P Auffinger & E Westhof. *Calculations of nucleic acid conformations.* Curr Opin Struct Biol, vol. 6, p 289–98, 1996.

[Lowe97] TM Lowe & SR Eddy. *tRNAscan-SE : a program for improved detection of transfer RNA genes in genomic sequence.* Nucleic Acids Res, vol. 25, p 955–64, 1997.

[Ludwig04] W Ludwig, O Strunk, R Westram, L Richter & H Meier. *ARB : a software environment for sequence data.* Nucleic Acids Res, vol. 32, no. 4, p 1363–71, 2004.

[Macke01] T. J. Macke, D. J. Ecker, R. R. Gutell, D. Gautheret, D. A. Case & R. Sampath. *RNAMotif, an RNA secondary structure definition and search algorithm.* Nucleic Acids Res, vol. 29, no. 22, p 4724–4735, 2001.

[Mangalam02] H Mangalam. *The Bio* toolkits–a brief overview.* Brief Bioinform, vol. 3, no. 3, p 296–302, 2002.

[Martinez08] HM Martinez, JV Jr Maizel & BA Shapiro. *RNA2D3D : A program for Generating, Viewing, and Comparing 3-Dimensional Models of RNA.* J Biomol Struct Dyn, vol. 25, no. 6, p 669–84, 2008.

[Masquida05] B Masquida & E Westhof. *Modeling the architecture of structured RNAs within a modular and hierarchical framework*. Handbook of RNA Biochemistry, Hartmann RK, Bindereif A, Schön A, Westhof E, p 536–545, 2005.

[Massire98] C Massire & E Westhof. *MANIP : an interactive tool for modelling RNA.* J Mol Graph Model, vol. 16, no. 4-6, p 197–205, 255–7, 1998.

[Mathews99] D. H. Mathews, J. Sabina, M. Zuker & D. H. Turner. *Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure*. J Mol Biol, vol. 288, no. 5, p 911–940, 1999.

[Mathews02] D. H. Mathews & D. H. Turner. *Dynalign : an algorithm for finding the secondary structure common to two RNA sequences*. J Mol Biol, vol. 317, no. 2, p 191–203, 2002.

[Mathews04a] DH Mathews. *Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization.* RNA, vol. 10, no. 8, p 1178–90, 2004.

[Mathews04b] DH Mathews, MD Disney, JL Childs, SJ Schroeder, M Zuker & DH Turner. *Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure.* Proc Natl Acad Sci U S A, vol. 101, p 7287–92, 2004.

[Mathews05] DH Mathews. *Predicting a set of minimal free energy RNA secondary structures common to two sequences.* Bioinformatics, vol. 21, no. 10, p 2246–53, 2005.

[Mathews07] DH Mathews, DH Turner & M Zuker. *RNA secondary structure prediction.* Curr Protoc Nucleic Acid Chem, vol. Chapter 11, p Unit 11.2, 2007.

[Matsui05] H. Matsui, K. Sato & Y. Sakakibara. *Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures*. Bioinformatics, vol. 21, no. 11, p 2611–2617, 2005.

[Matsumura03] S Matsumura, Y Ikawa & T Inoue. *Biochemical characterization of the kink-turn RNA motif.* Nucleic Acids Res, vol. 31, no. 19, p 5544–51, 2003.

[McCaskill90] JS McCaskill. *The equilibrium partition function and base pair binding probabilities for RNA secondary structure.* Biopolymers, vol. 29, p 1105–19, 1990.

[McCutcheon03] JP McCutcheon & SR Eddy. *Computational identification of non-coding RNAs in Saccharomyces cerevisiae by comparative genomics.* Nucleic Acids Res, vol. 31, p 4119–28, 2003.

[McDowall88] RD McDowall, JC Pearce & GS Murkitt. *Laboratory information management systems - part I. Concepts.* J Pharm Biomed Anal, vol. 6, no. 4, p 339–59, 1988.

[Mehler07] MF Mehler & JS Mattick. *Noncoding RNAs and RNA editing in brain development, functional diversification, and neurological disease.* Physiol Rev, vol. 87, p 799–823, 2007.

[Merelli07] E Merelli, G Armano, N Cannata, F Corradini, M d'Inverno, A Doms, P Lord, A Martin, L Milanesi, S Möller, M Schroeder & M Luck. *Agents in bioinformatics, computational and systems biology.* Brief Bioinform, vol. 8, no. 1, p 45–59, 2007.

[Michel90] F Michel & E Westhof. *Modelling of the three-dimensional architecture of group I catalytic introns based on comparative sequence analysis.* J Mol Biol, vol. 216, p 585–610, 1990.

[Mokdad06] A. Mokdad & N. B. Leontis. *Ribostral : an RNA 3D alignment analyzer and viewer based on basepair isostericities*. Bioinformatics, vol. 22, no. 17, p 2168–2170, 2006.

[Montgomery05] SB Montgomery, T Fu, J Guan, K Lin & SJ Jones. *An application of peer-to-peer technology to the discovery, use and assessment of bioinformatics programs.* Nat Methods, vol. 2, no. 8, p 563, 2005.

[Murray03] LJ Murray, WB 3rd Arendall, DC Richardson & JS Richardson. *RNA backbone is rotameric.* Proc Natl Acad Sci U S A, vol. 100, p 13904–9, 2003.

[Murray05] LJ Murray, JS Richardson, WB Arendall & DC Richardson. *RNA backbone rotamers–finding your way in seven dimensions.* Biochem Soc Trans, vol. 33, no. Pt 3, p 485–7, 2005.

[Nagaswamy02] U Nagaswamy, M Larios-Sanz, J Hury, S Collins, Z Zhang, Q Zhao & GE Fox. *NCIR : a database of non-canonical interactions in known RNA structures.* Nucleic Acids Res, vol. 30, no. 1, p 395–7, 2002.

[Namy06] O Namy, SJ Moran, DI Stuart, RJ Gilbert & I Brierley. *A mechanical explanation of RNA pseudoknot function in programmed ribosomal frameshifting.* Nature, vol. 441, p 244–7, 2006.

[Navas-Delgado06] I Navas-Delgado, Mdel M Rojano-Muñoz, S Ramírez, AJ Pérez, E Andrés León, JF Aldana-Montes & O Trelles. *Intelligent client for integrating bioinformatics services.* Bioinformatics, vol. 22, no. 1, p 106–11, 2006.

[ncbitoolkit] `http://www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC`.

[ncir] `http://prion.bchs.uh.edu/bp_type`.

[ncrna] `http://www.ncnra.org`.

[Neerincx05] PB Neerincx & JA Leunissen. *Evolution of web services in bioinformatics.* Brief Bioinform, vol. 6, no. 2, p 178–88, 2005.

[Nevskaya05] N Nevskaya, S Tishchenko, A Gabdoulkhakov, E Nikonova, O Nikonov, A Nikulin, O Platonova, M Garber, S Nikonov & W Piendl. *Ribosomal protein L1 recognizes the same specific structural motif in its target sites on the autoregulatory mRNA and 23S rRNA.* Nucleic Acids Res, vol. 33, no. -Binding Proteins, p 478–85, 2005.

[Nussinov78] R Nussinov, G Piecznik, JR Griggs & DJ Kleitman. *Algorithms for loop matching*. SIAM J Appl Math, vol. 35, p 68–82, 1978.

[Oinn04] T Oinn, M Addis, J Ferris, D Marvin, M Senger, M Greenwood, T Carver, K Glover, MR Pocock, A Wipat & P Li. *Taverna : a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, vol. 20, no. 17, p 3045–54, 2004.

[opencybele] `http://www.opencybele.org`.

[opengl] `http://www.opengl.org`.

[openqbs] `http://www.ebi.ac.uk/~senger/openbqs`.

[Pang05] KC Pang, S Stephen, PG Engström, K Tajul-Arifin, W Chen, C Wahlestedt, B Lenhard, Y Hayashizaki & JS Mattick. *RNAdb–a comprehensive mammalian noncoding RNA database.* Nucleic Acids Res, vol. 33, p D125–30, 2005.

[Pang07] KC Pang, S Stephen, ME Dinger, PG Engström, B Lenhard & JS Mattick. *RNAdb 2.0–an expanded database of mammalian non-coding RNAs.* Nucleic Acids Res, vol. 35, p D178–82, 2007.

[Parisien08] M Parisien & F Major. *The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data.* Nature, vol. 452, no. 7183, p 51–5, 2008.

[pathport] `http://pathport.vbi.vt.edu`.

[Pavesi04] G. Pavesi, G. Mauri, M. Stefani & G. Pesole. *RNAProfile : an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences*. Nucleic Acids Res, vol. 32, no. 10, p 3258–3269, 2004.

[Pitt01] WR Pitt, MA Williams, M Steven, B Sweeney, AJ Bleasby & DS Moss. *The Bioinformatics Template Library–generic components for biocomputing.* Bioinformatics, vol. 17, no. 8, p 729–37, 2001.

[Pley94] HW Pley, KM Flaherty & DB McKay. *Model for an RNA tertiary interaction from the structure of an intermolecular complex between a GAAA tetraloop and an RNA helix.* Nature, vol. 372, p 111–3, 1994.

[Popenda08] M Popenda, M Blazewicz, M Szachniuk & RW Adamiak. *RNA FRABASE version 1.0 : an engine with a database to search for the three-dimensional fragments within RNA structures.* Nucleic Acids Res, vol. 36, no. Database issue, p D386–91, 2008.

[pubmed] `http://www.ncbi.nlm.nih.gov/pubmed`.

[pymol] `http://pymol.sourceforge.net`.

[Pütz07] J Pütz, B Dupuis, M Sissler & C Florentz. *Mamit-tRNA, a database of mammalian mitochondrial tRNA primary and secondary structures.* RNA, vol. 13, p 1184–90, 2007.

[Reeder05] J. Reeder & R. Giegerich. *Consensus shapes : an alternative to the Sankoff algorithm for RNA consensus structure prediction*. Bioinformatics, vol. 21, no. 17, p 3516–3523, 2005.

[Reeder07a] J. Reeder, J. Reeder & R. Giegerich. *Locomotif : from graphical motif description to RNA motif search*. Bioinformatics, vol. 23, no. 13, p i392–400, 2007.

[Reeder07b] J. Reeder, P. Steffen & R. Giegerich. *pknotsRG : RNA pseudoknot folding including near-optimal structures and sliding windows*. Nucleic Acids Res, vol. 35, no. Web Server issue, p W320–4, 2007.

[Ren05] J. Ren, B. Rastegari, A. Condon & H. H. Hoos. *HotKnots : heuristic prediction of RNA secondary structures including pseudoknots*. Rna, vol. 11, no. 10, p 1494–1504, 2005.

[Ren08] LH Ren, YS Ding, YZ Shen & XF Zhang. *Multi-agent-based bio-network for systems biology : protein-protein interaction network as an example.* Amino Acids, 2008.

[Rice00] P Rice, I Longden & A Bleasby. *EMBOSS : the European Molecular Biology Open Software Suite.* Trends Genet, vol. 16, no. 6, p 276–7, 2000.

[Richardson08] JS Richardson, B Schneider, LW Murray, GJ Kapral, RM Immormino, JJ Headd, DC Richardson, D Ham, E Hershkovits, LD Williams, KS Keating, AM Pyle, D Micallef, J Westbrook & HM Berman. *RNA backbone : consensus all-angle conformers and modular string nomenclature (an RNA Ontology Consortium contribution).* RNA, vol. 14, no. 3, p 465–81, 2008.

[Rietveld82] K Rietveld, R Van Poelgeest, CW Pleij, JH Van Boom & L Bosch. *The tRNA-like structure at the 3' terminus of turnip yellow mosaic virus RNA. Differences and similarities with canonical tRNA.* Nucleic Acids Res, vol. 10, p 1929–46, 1982.

[Rivas99] E Rivas & SR Eddy. A dynamic programming algorithm for rna structure prediction including pseudoknots., volume 285. Reverse Transcriptase, 1999.

[Rivas01a] E. Rivas & S. R. Eddy. *Noncoding RNA gene detection using comparative sequence analysis*. BMC Bioinformatics, vol. 2, p 8, 2001.

[Rivas01b] E Rivas, RJ Klein, TA Jones & SR Eddy. *Computational identification of noncoding RNAs in E. coli by comparative genomics.* Curr Biol, vol. 11, p 1369–73, 2001.

[Robinson83] AL Robinson. *LIMS Is Next Step in Laboratory Automation.* Science, vol. 220, no. 4593, p 180–183, 1983.

[Romby06] P Romby, F Vandenesch & EG Wagner. *The role of RNAs in the regulation of virulence-gene expression.* Curr Opin Microbiol, vol. 9, p 229–36, 2006.

[Rozhdestvensky03] TS Rozhdestvensky, TH Tang, IV Tchirkova, J Brosius, JP Bachellerie & A Hüttenhofer. *Binding of L7Ae protein to the K-turn of archaeal snoRNAs : a shared RNA binding motif for C/D and H/ACA box snoRNAs in Archaea.* Nucleic Acids Res, vol. 31, no. -Binding Proteins, p 869–77, 2003.

[Ruan04] J Ruan, GD Stormo & W Zhang. *ILM : a web server for predicting RNA secondary structures with pseudoknots.* Nucleic Acids Res, vol. 32, no. Web Server issue, p W146–9, 2004.

[Rázga04] F Rázga, N Spackova, K Réblova, J Koca, NB Leontis & J Sponer. *Ribosomal RNA kink-turn motif–a flexible molecular hinge.* J Biomol Struct Dyn, vol. 22, p 183–94, 2004.

[Saenger84] W Saenger. Principles of nucleic acid structure. Spring-Verlag, 1984.

[Sankoff85] D Sankoff. *Simultaneous solution of the RNA folding, alignment and protosequence problems*. J.Appl.Math, vol. 45, p 810–825, 1985.

[Sarver08] M Sarver, CL Zirbel, J Stombaugh, A Mokdad & NB Leontis. *FR3D : finding local and composite recurrent structural motifs in RNA 3D structures.* J Math Biol, vol. 56, no. 1-2, p 215–52, 2008.

[Sato05] K. Sato & Y. Sakakibara. *RNA secondary structural alignment with conditional random fields*. Bioinformatics, vol. 21 Suppl 2, p ii237–ii242, 2005.

[Schatz07] MC Schatz, C Trapnell, AL Delcher & A Varshney. *High-throughput sequence alignment using Graphics Processing Units.* BMC Bioinformatics, vol. 8, no. 1, p 474, 2007.

[Schluenzen00] F Schluenzen, A Tocilj, R Zarivach, J Harms, M Gluehmann, D Janell, A Bashan, H Bartels, I Agmon, F Franceschi & A Yonath. *Structure of functionally activated small ribosomal subunit at 3.3 angstroms resolution.* Cell, vol. 102, p 615–23, 2000.

[Schneider04] B Schneider, Z Morávek & HM Berman. *RNA conformational classes.* Nucleic Acids Res, vol. 32, p 1666–77, 2004.

[Schwieters03] CD Schwieters, JJ Kuszewski, N Tjandra & GM Clore. *The Xplor-NIH NMR molecular structure determination package.* J Magn Reson, vol. 160, no. 1, p 65–73, 2003.

[Seibel06] P. N. Seibel, T. Muller, T. Dandekar, J. Schultz & M. Wolf. *4SALE–a tool for synchronous RNA sequence and secondary structure alignment and editing*. BMC Bioinformatics, vol. 7, p 498, 2006.

[Senger03] S Senger. *SOAPLAB – a unified sesame door to analysis tools*. Proceedings of the UK e-Science All Hands Meeting, p 509–513, 2003.

[Shafaei05] Sima Shafaei & Nasser Ghasem Aghaee. *Biological Network Simulation Using Holonic Multiagent Systems*. Proceedings in the 10th Int. Conf. on Computer Modeling and Simulation, p 617–622, 2005.

[Shah04] SP Shah, DY He, JN Sawkins, JC Druce, G Quon, D Lett, GX Zheng, T Xu & BF Ouellette. *Pegasys : software for executing and integrating analyses of biological sequences.* BMC Bioinformatics, vol. 5, p 40, 2004.

[Shahi06] P Shahi, S Loukianiouk, A Bohne-Lang, M Kenzelmann, S Küffer, S Maertens, R Eils, HJ Gröne, N Gretz & B Brors. *Argonaute–a database for gene regulation by mammalian microRNAs.* Nucleic Acids Res, vol. 34, no. Database issue, p D115–8, 2006.

[Shannon03] P Shannon, A Markiel, O Ozier, NS Baliga, JT Wang, D Ramage, N Amin, B Schwikowski & T Ideker. *Cytoscape : a software environment for integrated models of biomolecular interaction networks.* Genome Res, vol. 13, no. 11, p 2498–504, 2003.

[Shapiro96] BA Shapiro & W Kasprzak. *STRUCTURELAB : a heterogeneous bioinformatics system for RNA structure analysis.* J Mol Graph, vol. 14, p 194–205, 222–4, 1996.

[Shapiro01] BA Shapiro, JC Wu, D Bengali & MJ Potts. *The massively parallel genetic algorithm for RNA folding : MIMD implementation and population variation.* Bioinformatics, vol. 17, no. 2, p 137–48, 2001.

[Shapiro06] BA Shapiro, W Kasprzak, C Grunewald & J Aman. Graphical exploratory data analysis of rna secondary structure dynamics predicted by the massively parallel genetic algorithm., volume 25. -1, 2006.

[Shapiro07] BA Shapiro, YG Yingling, W Kasprzak & E Bindewald. *Bridging the gap in RNA structure prediction.* Curr Opin Struct Biol, vol. 17, no. 2, p 157–65, 2007.

[Shefer07] K Shefer, Y Brown, V Gorkovoy, T Nussbaum, NB Ulyanov & Y Tzfati. *A triple helix within a pseudoknot is a conserved and essential element of telomerase RNA.* Mol Cell Biol, vol. 27, no. 6, p 2130–43, 2007.

[Shindyalov98] IN Shindyalov & PE Bourne. *Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.* Protein Eng, vol. 11, no. 9, p 739–47, 1998.

[Siebert05] S. Siebert & R. Backofen. *MARNA : multiple alignment and consensus structure prediction of RNAs based on sequence structure comparisons*. Bioinformatics, vol. 21, no. 16, p 3352–3359, 2005.

[Simons97] KT Simons, C Kooperberg, E Huang & D Baker. *Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions.* J Mol Biol, vol. 268, no. 1, p 209–25, 1997.

[Smit08] S Smit, K Rother, J Heringa & R Knight. *From knotted to nested RNA structures : a variety of computational methods for pseudoknot removal.* RNA, vol. 14, no. 3, p 410–6, 2008.

[Smith05] B Smith, W Ceusters, B Klagges, J Köhler, A Kumar, J Lomax, C Mungall, F Neuhaus, AL Rector & C Rosse. *Relations in biomedical ontologies.* Genome Biol, vol. 6, no. 5, p R46, 2005.

[Smith07] B Smith, M Ashburner, C Rosse, J Bard, W Bug, W Ceusters, LJ Goldberg, K Eilbeck, A Ireland & CJ Mungall. *The OBO Foundry : coordinated evolution of ontologies to support biomedical data integration.* Nat Biotechnol, vol. 25, no. 11, p 1251–5, 2007.

[soaplab] `http://www.ebi.ac.uk/soaplab`.

[Song06] Y Song, C Liu, RL Malmberg, C He & L Cai. *Memory efficient alignment between RNA sequences and stochastic grammar models of pseudoknots.* Int J Bioinform Res Appl, vol. 2, p 289–304, 2006.

[Sperschneider08] J Sperschneider & A Datta. *KnotSeeker : Heuristic pseudoknot detection in long RNA sequences.* RNA, 2008.

[Spjuth07] O. Spjuth, T. Helmus, E. L. Willighagen, S. Kuhn, M. Eklund, J. Wagener, P. Murray-Rust, C. Steinbeck & J. E. Wikberg. *Bioclipse : an open source workbench for chemo- and bioinformatics*. BMC Bioinformatics, vol. 8, p 59, 2007.

[ssprediction] `http://en.wikipedia.org/wiki/List_of_RNA_structure_prediction_software`.

[St-Onge07] K St-Onge, P Thibault, S Hamel & F Major. *Modeling RNA tertiary structure motifs by graph-grammars.* Nucleic Acids Res, vol. 35, p 1726–36, 2007.

[Stajich02] JE Stajich, D Block, K Boulez, SE Brenner, SA Chervitz, C Dagdigian, G Fuellen, JG Gilbert, I Korf, H Lapp, H Lehväslaiho, C Matsalla, CJ Mungall, BI Osborne, MR Pocock, P Schattner, M Senger, LD Stein, E Stupka, MD Wilkinson & E Birney. *The Bioperl toolkit : Perl modules for the life sciences.* Genome Res, vol. 12, no. 10, p 1611–8, 2002.

[Staple05] DW Staple, SE Butcher & S Virus. *Pseudoknots : RNA structures with diverse functions.* PLoS Biol, vol. 3, p e213, 2005.

[Steffen06] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder & R. Giegerich. *RNAshapes : an integrated RNA analysis package based on abstract shapes*. Bioinformatics, vol. 22, no. 4, p 500–503, 2006.

[Stein02] Lincoln D Stein. *Creating a bioinformatics nation*. Nature, vol. 417, p 119–120, 2002.

[Stevens03] RD Stevens, AJ Robinson & CA Goble. *myGrid : personalised bioinformatics on the information grid.* Bioinformatics, vol. 19 Suppl 1, p i302–4, 2003.

[Strobel04] SA Strobel, PL Adams, MR Stahley & J Wang. *RNA kink turns to the left and to the right.* RNA, vol. 10, p 1852–4, 2004.

[Sun07] Y. Sun, S. Zhao, H. Yu, G. Gao & J. Luo. *ABCGrid : Application for Bioinformatics Computing Grid*. Bioinformatics, vol. 23, no. 9, p 1175–1177, 2007.

[Swertz07] MA Swertz & RC Jansen. *Beyond standardization : dynamic software infrastructures for systems biology.* Nat Rev Genet, vol. 8, no. 3, p 235–43, 2007.

[Szymanski07] M Szymanski, VA Erdmann & J Barciszewski. *Noncoding RNAs database (ncRNAdb).* Nucleic Acids Res, vol. 35, p D162–4, 2007.

[Tabei06] Y Tabei, K Tsuda, T Kin & K Asai. *SCARNA : fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments.* Bioinformatics, vol. 22, no. 14, p 1723–9, 2006.

[Tabei08] Y Tabei, H Kiryu, T Kin & K Asai. *A fast structural multiple alignment method for long RNA sequences.* BMC Bioinformatics, vol. 9, no. 1, p 33, 2008.

[Taft07] RJ Taft, M Pheasant & JS Mattick. *The relationship between non-protein-coding DNA and eukaryotic complexity.* Bioessays, vol. 29, p 288–99, 2007.

[Tamura04] M. Tamura, D. K. Hendrix, P. S. Klosterman, N. R. Schimmelman, S. E. Brenner & S. R. Holbrook. *SCOR : Structural Classification of RNA, version 2.0*. Nucleic Acids Res, vol. 32, no. Database issue, p D182–4, 2004.

[Tang05] F Tang, CL Chua, LY Ho, YP Lim, P Issac & A Krishnan. *Wildfire : distributed, Grid-enabled workflow construction and execution.* BMC Bioinformatics, vol. 6, p 69, 2005.

[Thebault06] P. Thebault, S. de Givry, T. Schiex & C. Gaspin. *Searching RNA motifs and their intermolecular contacts with constraint networks*. Bioinformatics, vol. 22, no. 17, p 2074–2080, 2006.

[Thompson94] JD Thompson, DG Higgins & TJ Gibson. *CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.* Nucleic Acids Res, vol. 22, no. 22, p 4673–80, 1994.

[tinker] `http://dasher.wustl.edu/tinker`.

[Torarinsson07] E. Torarinsson, J. H. Havgaard & J. Gorodkin. *Multiple structural alignment and clustering of RNA sequences*. Bioinformatics, vol. 23, no. 8, p 926–932, 2007.

[Torres-Larios02] A Torres-Larios, AC Dock-Bregeon, P Romby, B Rees, R Sankaranarayanan, J Caillet, M Springer, C Ehresmann, B Ehresmann & D Moras. *Structural basis of translational control by Escherichia coli threonyl tRNA synthetase.* Nat Struct Biol, vol. 9, p 343–7, 2002.

[Touzet04] H. Touzet & O. Perriquet. *CARNAC : folding families of related RNAs*. Nucleic Acids Res, vol. 32, no. Web Server issue, p W142–5, 2004.

[Touzet07] H Touzet. *Comparative analysis of RNA genes : the caRNAc software.* Methods Mol Biol, vol. 395, p 465–74, 2007.

[U1snRNA] `http://www-ibmc.u-strasbg.fr/arn/Westhof/fig_them3_West/u1.pdb`.

[Ulyanov07] NB Ulyanov, K Shefer, TL James & Y Tzfati. *Pseudoknot structures with conserved base triples in telomerase RNAs of ciliates.* Nucleic Acids Res, vol. 35, p 6150–60, 2007.

[van Batenburg95] FH van Batenburg, AP Gultyaev & CW Pleij. *An APL-programmed genetic algorithm for the prediction of RNA secondary structure.* J Theor Biol, vol. 174, no. 3, p 269–80, 1995.

[van Batenburg00] FH van Batenburg, AP Gultyaev, CW Pleij, J Ng & J Oliehoek. *PseudoBase : a database with RNA pseudoknots.* Nucleic Acids Res, vol. 28, no. 1, p 201–4, 2000.

[van Batenburg01] FH van Batenburg, AP Gultyaev & CW Pleij. *PseudoBase : structural information on RNA pseudoknots.* Nucleic Acids Res, vol. 29, no. 1, p 194–5, 2001.

[Van Walle04] I. Van Walle, I. Lasters & L. Wyns. *Align-m–a new algorithm for multiple alignment of highly divergent sequences*. Bioinformatics, vol. 20, no. 9, p 1428–1435, 2004.

[Vidovic00] I Vidovic, S Nottrott, K Hartmuth, R Lührmann & R Ficner. *Crystal structure of the spliceosomal 15.5kD protein bound to a U4 snRNA fragment.* Mol Cell, vol. 6, no. -Binding Proteins, p 1331–42, 2000.

[Vignal96] L Vignal, Y d'Aubenton Carafa, F Lisacek, E Mephu Ngüifo, P Rouzé, J Quinqueton & C Thermes. *Exon prediction in eucaryotic genomes.* Biochimie, vol. 78, p 327–34, 1996.

[Vignal97] L Vignal & F Lisacek. *A Multi-Agent System for Exon Prediction in Human Sequences.* Genome Inform Ser Workshop Genome Inform, vol. 8, p 156–165, 1997.

[Vignal99] L Vignal, F Lisacek, J Quinqueton, Y d'Aubenton Carafa & C Thermes. *A multi-agent system simulating human splice site recognition.* Comput Chem, vol. 23, no. Splicing, p 219–31, 1999.

[Voss06] B. Voss. *Structural analysis of aligned RNAs*. Nucleic Acids Res, vol. 34, no. 19, p 5471–5481, 2006.

[wade] `http://jade.tilab.com/wade`.

[Wadley04] LM Wadley & AM Pyle. *The identification of novel RNA structural motifs using COMPADRES : an automated approach to structural discovery.* Nucleic Acids Res, vol. 32, no. 22, p 6650–9, 2004.

[Wang05] X. Wang, J. Zhang, F. Li, J. Gu, T. He, X. Zhang & Y. Li. *MicroRNA identification based on sequence and structure alignment*. Bioinformatics, vol. 21, no. 18, p 3610–3614, 2005.

[Wang08] X Wang. *miRDB : A microRNA target prediction and functional annotation database with a wiki interface.* RNA, 2008.

[Washietl05] S. Washietl, I. L. Hofacker & P. F. Stadler. *Fast and reliable prediction of noncoding RNAs*. Proc Natl Acad Sci U S A, vol. 102, no. 7, p 2454–2459, 2005.

[Waugh02] A Waugh, P Gendron, R Altman, JW Brown, D Case, D Gautheret, SC Harvey, N Leontis, J Westbrook, E Westhof, M Zuker & F Major. *RNAML : a standard syntax for exchanging RNA information.* RNA, vol. 8, no. 6, p 707–17, 2002.

[Weinberg06] Z. Weinberg & W. L. Ruzzo. *Sequence-based heuristics for faster annotation of non-coding RNA families*. Bioinformatics, vol. 22, no. 1, p 35–39, 2006.

[Westhof85] E Westhof, P Dumas & D Moras. *Crystallographic refinement of yeast aspartic acid transfer RNA*. J Mol Biol, vol. 184, p 119–45, 1985.

[Westhof89] E Westhof, P Romby, PJ Romaniuk, JP Ebel, C Ehresmann & B Ehresmann. *Computer modeling from solution data of spinach chloroplast and of Xenopus laevis somatic and oocyte 5 S rRNAs.* J Mol Biol, vol. 207, p 417–31, 1989.

[Westhof90] E Westhof, P Romby, C Ehresmann & B Ehresmann. *Computer-Aided Structural Biochemistry*. Theoretical Biochemistry & Molecular Biophysics, vol. Adenine Press, p 399–409, 1990.

[Westhof93] Eric Westhof. *Modelling the three-dimensional structure of ribonucleic acids*. J. Mol. Struct., vol. 286, p 203–210, 1993.

[Westhof02] E Westhof. *Group I introns and RNA folding.* Biochem Soc Trans, vol. 30, p 1149–52, 2002.

[Westhof04] E Westhof & C Massire. *Structural biology. Evolution of RNA architecture.* Science, vol. 306, p 62–3, 2004.

[Wexler07] Y. Wexler, C. Zilberstein & M. Ziv-Ukelson. *A study of accessible motifs and RNA folding complexity*. J Comput Biol, vol. 14, no. 6, p 856–872, 2007.

[Wheeler07] T. J. Wheeler & J. D. Kececioglu. *Multiple alignment by aligning alignments*. Bioinformatics, vol. 23, no. 13, p i559–68, 2007.

[Wiese05] KC Wiese, E Glen & A Vasudevan. *JViz.Rna–a Java tool for RNA secondary structure visualization.* IEEE Trans Nanobioscience, vol. 4, no. 3, p 212–8, 2005.

[Wilkinson02] MD Wilkinson & M Links. *BioMOBY : an open source biological web services proposal.* Brief Bioinform, vol. 3, no. 4, p 331–41, 2002.

[Wilkinson08] MD Wilkinson, M Senger, E Kawas, R Bruskiewich, J Gouzy, C Noirot, P Bardou, A Ng, D Haase, A Saiz Ede, D Wang, F Gibbons, PM Gordon, CW Sensen, JM Carrasco, JM Fernández, L Shen, M Links, M Ng, N Opushneva, PB Neerincx, JA Leunissen, R Ernst, S Twigger, B Usadel, B Good, Y Wong, L Stein, W Crosby, J Karlsson, R Royo, I Párraga, S Ramírez, JL Gelpi, O Trelles, DG Pisano, N Jimenez, A Kerhornou, R Rosset, L Zamacola, J Tarraga, J Huerta-Cepas, JM Carazo, J Dopazo, R Guigo, A Navarro, M Orozco, A Valencia, MG Claros, AJ Pérez, J Aldana, MM Rojano, R Fernandez-Santa Cruz, I Navas, G Schiltz, A Farmer, r D Gessle, H Schoof & A Groscurth. *Interoperability with Moby 1.0– It's better than sharing your toothbrush !* Brief Bioinform, 2008.

[Wilm08] A Wilm, K Linnenbrink & G Steger. *ConStruct : improved construction of RNA consensus structures.* BMC Bioinformatics, vol. 9, no. 1, p 219, 2008.

[Wimberly00] BT Wimberly, DE Brodersen, WM Jr Clemons, RJ Morgan-Warren, AP Carter, C Vonrhein, T Hartsch & V Ramakrishnan. *Structure of the 30S ribosomal subunit.* Nature, vol. 407, p 327–39, 2000.

[Witwer04] C Witwer, IL Hofacker & PF Stadler. *Prediction of consensus RNA secondary structures including pseudoknots.* IEEE/ACM Trans Comput Biol Bioinform, vol. 1, p 66–77, 2004.

[Wuchty99] S Wuchty, W Fontana, IL Hofacker & P Schuster. *Complete suboptimal folding of RNA and the stability of secondary structures.* Biopolymers, vol. 49, p 145–65, 1999.

[Xayaphoummine03] A Xayaphoummine, T Bucher, F Thalmann & H Isambert. *Prediction and statistics of pseudoknots in RNA structures using exactly clustered stochastic simulations.* Proc Natl Acad Sci U S A, vol. 100, no. 26, p 15310–5, 2003.

[Xayaphoummine05] A. Xayaphoummine, T. Bucher & H. Isambert. *Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots*. Nucleic Acids Res, vol. 33, no. Web Server issue, p W605–10, 2005.

[xith3d] `http://xith.org`.

[xmlcentral] `http://xml.nig.ac.jp`.

[Xu07] X. Xu, Y. Ji & G. D. Stormo. *RNA Sampler : a new sampling based algorithm for common RNA secondary structure prediction and structural alignment*. Bioinformatics, vol. 23, no. 15, p 1883–1891, 2007.

[Yang03] H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H. Berman & E. Westhof. *Tools for the automatic identification and classification of RNA base pairs*. Nucleic Acids Res, vol. 31, no. 13, p 3450–3460, 2003.

[Yang04] Q. Yang & M. Blanchette. *StructMiner : a tool for alignment and detection of conserved secondary structure*. Genome Inform, vol. 15, no. 2, p 102–111, 2004.

[Yao06] Z. Yao, Z. Weinberg & W. L. Ruzzo. *CMfinder–a covariance model based RNA motif finding algorithm*. Bioinformatics, vol. 22, no. 4, p 445–452, 2006.

[Ying04] X. Ying, H. Luo, J. Luo & W. Li. *RDfolder : a web server for prediction of RNA secondary structure*. Nucleic Acids Res, vol. 32, no. Web Server issue, p W150–3, 2004.

[Zhang08] L Zhang, N Yang & G Coukos. *MicroRNA in human cancer : one step forward in diagnosis and treatment.* Adv Exp Med Biol, vol. 622, p 69–78, 2008.

[Zuker03] M. Zuker. *Mfold web server for nucleic acid folding and hybridization prediction*. Nucleic Acids Res, vol. 31, no. 13, p 3406–3415, 2003.