Prédiction de structure tridimensionnelle de molécules d'ARN par minimisation de regret

Mélanie Boudard

NNT : 2016SACLV026

Thèse de Doctorat
de
l'Université Paris-Saclay
préparée à
l'Université Versailles Saint-Quentin en Yvelines
et à
l'Université Paris Sud

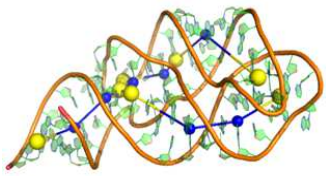
ÉCOLE DOCTORALE N°580
STIC : Sciences et technologies de l'information et de la communication

Spécialité de doctorat : Informatique

Par

Mlle Mélanie Boudard

Prédiction de structure tridimensionnelle de molécules d'ARN par minimisation de regret



Thèse présentée et soutenue à Gif-sur-Yvette, le 29 avril 2016 :

Composition du Jury :
Mme Alessandra Carbone, Professeure, UPMC, Présidente du jury
M. Christoph Dürr, Directeur de recherche, CNRS, Rapporteur
M. Jérôme Waldispühl, Professeur, McGill University, Rapporteur
M. Raphaël Guérois, Directeur de recherche, CEA, Examinateur
M. Stefano Moretti, Chargé de recherche, CNRS, Examinateur
Mme Johanne Cohen, Chargée de recherche, CNRS, Directrice de thèse
M. Alain Denise, Professeur, UPSud, Directeur de thèse
M. Dominique Barth, Professeur, UVSQ, Directeur de thèse

Remerciements

Une thèse, c'est une aventure, souvent semée d'embûches, durant laquelle nous pouvons croiser toutes sortes de personnes. Je vais sûrement en oublier, mais je souhaite remercier ceux qui ont aidé à faire cette thèse, que ce soit scientifiquement ou humainement.

Je remercie mes directeurs de thèse, Johanne Cohen, Alain Denise et Dominique Barth, d'avoir accepté de me confier ce travail alors que vous ne me connaissiez pas. Je vous remercie de m'avoir poussé et de m'avoir interrogé pour que je puisse donner le meilleur de moi-même. Je remercie tout particulièrement Johanne, avec qui j'ai partagé plus qu'un travail scientifique et avec qui j'ai pu évoluer durant ces années de recherche.
Je remercie Christoph Dürr et Jérôme Waldispühl d'avoir accepté d'être mes rapporteurs de thèse. Vos commentaires m'ont permis d'appréhender ma thèse avec un regard extérieur et d'apporter de nouvelles idées pour la suite.
Je remercie mon jury de thèse, Raphaël Guérois, Stefano Moretti et surtout la présidente, Alessandra Carbone, pour leurs questions et remarques pertinentes durant ma soutenance.
Je tiens aussi à remercier Julie Bernauer qui m'a apporté de nombreuses connaissances scientifiques nécessaires, et qui m'a permis d'avoir une ligne directrice pour ma thèse.
J'ai eu l'occasion de travailler dans deux laboratoires, le LRI et DAVID. J'ai ainsi pu rencontrer plusieurs équipes, pour lesquelles j'ai notamment enseigné, à Versailles comme à Paris-Sud. J'ai ainsi pu rencontrer de nombreux collègues, qui se reconnaîtront. En particulier Lise Rodier, avec qui j'ai partagé plus qu'une directrice de thèse.

Je remercie mes parents, mon frère, Sophie, et Véronique, qui m'ont soutenue depuis le début de mes études et jusqu'à maintenant. Je remercie et je pense fort Charles, qui m'a supportée tous les jours, durant les moments difficiles, et même durant les relectures de mes travaux. Je remercie Olivier Hermant pour m'avoir poussée à faire de la recherche, ainsi que Matthieu Manceny, Raja Chiky et tous les professeurs de l'ISEP.
Comme toute personne qui vient de vivre une aventure extraordinaire, je remercie ceux qui ont été là, qui m'ont permis d'avancer, de souffler, de réfléchir et de rire : Romain, Charlotte, Maxime, Ambroise, Benoit, Charles, Thomas, Nelson, Emmanuelle, Stan, Paul, Lucie, les Caves Alliées, ... À tous ceux avec qui j'ai partagé un café, un verre ou une discussion. À tous mes collègues, mes professeurs et mes élèves.
Merci à ceux qui ont participé à cette thèse et qui ont vécu avec moi ces années que je n'oublierai jamais !

Résumé

Les fonctions d'une molécule d'ARN sont très étroitement liées à sa structure tridimensionnelle. Il est essentiel de pouvoir prédire cette structure pour étudier sa fonction. Le repliement de l'ARN dans un espace 3D peut être vu comme un processus en deux étapes : (i) le repliement en structure secondaire, grâce à des interactions fortes, (ii) le repliement en structure tridimensionnelle par des interactions tertiaires. Prédire la structure secondaire a donné lieu à de nombreuses avancées depuis plus de trente ans. Toutefois, la prédiction de la structure tridimensionnelle est difficile. Nous nous intéressons ici au problème de prédiction de la structure 3D d'ARN. L'objectif est de développer une méthode de repliement d'ARN fonctionnant sur tous les types de molécule d'ARN et obtenant des structures similaires aux molécules réelles. Dans ce document, nous représentons la structure secondaire de l'ARN comme un graphe. Cette modélisation permet de réaliser un repliement global dans l'espace. Notre paradigme consiste à voir la structure 3D comme un équilibre en théorie des jeux. Nous formalisons donc un jeu en nous basant notamment sur des statistiques. Pour atteindre un équilibre, nous utiliserons des algorithmes de minimisation de regret. Notre méthode, nommée GARN, utilise des techniques originales pour ce problème (comme la minimisation de regret). Elle nous a permis d'extraire des paramètres importants pour une bonne prédiction à gros grain de la structure de l'ARN.

Mots clefs : ARN, structure 3D, théorie des jeux, minimisation de regret, potentiel statistique.

Table des matières

Introduction

La découverte de la structure en double hélice de l'ADN par Watson et Crick en 1953 [Watson et Crick, 1953] peut être considérée comme le début de la biologie moléculaire telle que nous la connaissons actuellement. Support des gènes, l'ADN possède cette structure pratiquement universelle et fait l'objet de très nombreuses études depuis sa découverte. Cependant, l'ADN est loin d'être le seul acide nucléique digne d'intérêt. Dans le déroulement classique de l'expression d'un gène, l'ARN (Acide RiboNucléique) est le *messager* entre l'ADN et la protéine [Gros *et al.*, 1961, Jacob et Monod, 1961]. De nombreux autres rôles leur ont été attribués, de la traduction de l'ARN messager à la régulation de l'expression du génome cellulaire. La difficulté d'étude de ces ARN est due principalement à leur instabilité relative (de quelques minutes à quelques heures [Parker et Song, 2004]). En effet, la nature monocaténaire (simple brin) de l'ARN l'empêche de se stabiliser efficacement et il se dégrade donc rapidement. Néanmoins, cette instabilité relative est nécessaire à la plupart des fonctions de l'ARN.

Les ARN forment une famille avec de nombreux sous-genres et activités spécifiques. Outre l'ARN messager (noté ARNm), qui transmet le message du gène et qui sera traduit en protéine par le ribosome (complexe protéine-ARN), il existe de nombreux autres ARN dits non-codants. Les ARN ribosomaux (ARNr) constituent le ribosome. Ils ont pour tâches principales de maintenir la structure du ribosome et de fixer les ARNm pendant leur traduction. Les ARN de transfert (ARNt) vont faire correspondre un codon (un enchaînement de trois bases successives de l'ARNm) à un acide aminé spécifique. Les ARN interférents (ARNi) sont des ARN synthétisés en laboratoire. Ils permettent, lorsqu'ils sont injectés dans une cellule, de se fixer sur les ARNm à leur sortie du noyau et donc d'empêcher leur traduction en protéines. Cela occasionne une baisse importante de la quantité de cette protéine dans la cellule. Cette idée, datant de 1998 [Fire *et al.*, 1998], ne fut mise en application chez l'homme que suite à la découverte de Thomas Tuschl [Meister et Tuschl, 2004].

Les ARN viraux sont pour la plupart des ARN de grande taille, avec des séquences relativement variables. La prédiction du repliement de ces ARN pourrait permettre de créer des interactions entre eux et d'autres molécules, en se focalisant sur leur structure pour inhiber leurs effets.

L'ARN pourrait donc être utilisé à des fins médicales (tel que les ARNi), notamment dans la régulation de l'expression génétique, comme guide pour des enzymes, dans le contrôle de la réplication des plasmides, etc. Les molécules d'ARN sont donc impliquées dans de nombreux processus biologiques au sein des cellules. La compréhension de la forme 3D adoptée par les molécules d'ARN donne des informations nécessaires sur la fonction jouée par ces molécules. Cela permettrait une plus grande tolérance aux mutations des séquences cibles et donc des traitements plus efficaces. La connaissance de ces structures est essentielle pour le développement de méthodes thérapeutiques [Cooper *et al.*, 2009] ou dans des domaines émergents comme les nanotechnologies [Guo, 2010].

La structure de l'ARN est composée d'un alphabet plus simple que celui des protéines, mais la prédiction de structure 3D reste compliquée [Tinoco et Bustamante, 1999]. L'une des principales difficultés vient des modèles thermodynamiques réalistes qui rendent le problème de repliement NP-difficile si nous les utilisons tels quels [Sheikh *et al.*, 2012].

La nature hiérarchique du repliement d'ARN [Brion et Westhof, 1997, Batey *et al.*, 1999, Tinoco et Bustamante, 1999] permet de trouver des méthodes de modélisation en se basant sur des connaissances des structures intermédiaires, telle que les structures secondaires (repliement de Watson-Crick et *Wobble*). Il existe des approches pour prédire les structures secondaires [Zuker, 2003, Mathews, 2006, Reeder *et al.*, 2006, Shapiro *et al.*, 2007, Hofacker, 2009]. La connaissance de ces structures permet ainsi une première approche de la modélisation 3D, avec comme information de base un premier niveau de repliement en 2D.

Pour trouver la structure 3D, il existe des outils interactifs, tels que RNA2D3D [Martinez *et al.*, 2008] et Assemble [Jossinet *et al.*, 2010], qui permettent de construire les structures 3D grâce à des motifs réguliers. Ces méthodes demandent néanmoins la connaissance d'un expert pour la reconstruction de formes fiables.

Les méthodes actuelles pour la prédiction automatisée de structures 3D [Laing et Schlick, 2011, Rother *et al.*, 2011, Cruz *et al.*, 2012, Sim *et al.*, 2012b] se basent principalement sur de nombreuses études de classifications des interactions chimiques et biologiques [Leontis et Westhof, 2001, Murray *et al.*, 2003, Sykes et Levitt, 2005, Leontis *et al.*, 2006, Das et Baker, 2007, Frellsen *et al.*, 2009]. Ces approches automatisées peuvent aussi utiliser des bibliothèques de fragments [Das et Baker, 2007, Parisien et Major, 2008] ou des informations sur les paires de bases et les empilements [Dima *et al.*, 2005, Sharma *et al.*, 2008, Flores et Altman, 2010].

Ces méthodes de prédiction travaillent principalement au niveau du nucléotide [Das et Baker, 2007,Parisien et Major, 2008,Sharma *et al.*, 2008,Jonikas *et al.*, 2009,Boniecki *et al.*, 2015]. L'utilisation de nouveaux niveaux de représentation est l'une des caractéristiques majeures des nouvelles méthodes de prédiction. L'utilisation de la nature hiérarchique de l'ARN permet de travailler sur des modèles à gros grain simplifiant la représentation de la

molécule [Le *et al.*, 1989, Shapiro et Zhang, 1990, Laing et Schlick, 2011, Sim *et al.*, 2012a, Laing *et al.*, 2013, Lamiable *et al.*, 2013, Fonseca *et al.*, 2014, Kim *et al.*, 2014, Kerpedjiev *et al.*, 2015]. Ces représentations sont actuellement utilisées pour de l'échantillonnage des structures 3D par des techniques probabilistes, telles que les méthodes de Monte-Carlo [Metropolis et Ulam, 1949, Metropolis, 1987].

Dans ce document, nous avons travaillé à la conception d'une méthode globale [Boudard *et al.*, 2015] permettant d'échantillonner et de prédire la structure des molécules d'ARN. Les solutions actuellement utilisées travaillent au niveau du nucléotide. Le modèle à gros grain, en pleine émergence actuellement, obtient des résultats prometteurs. Il permet de simplifier la vision du repliement. Partant de cette idée, nous avons étudié la possibilité d'utiliser la théorie des jeux pour la recherche d'un repliement à gros grain. En effet, une première approche de la théorie des jeux pour l'ARN [Lamiable *et al.*, 2013] a donné des résultats probants. Ces résultats motivent l'idée de tester l'apport de la théorie des jeux, et plus spécifiquement les algorithmes de minimisation de regret, sur un modèle à gros grain. Notre objectif est d'échantillonner un espace de solutions de structures d'une molécule d'ARN. Dans nos échantillons, nous souhaitons qu'une ou plusieurs structures soient assez similaires à la structure réelle. Notre approche, GARN (*Game Algorithm for RNa*), simplifie la représentation de la molécule et de son repliement. GARN utilisera une représentation à gros grains pouvant se replier dans l'espace grâce à des algorithmes de minimisation de regret. Le repliement sera formalisé comme un jeu, avec des joueurs choisissant des stratégies pour recevoir des gains. Ce jeu possédera plusieurs paramètres modifiables pour s'adapter à la molécule. Ces paramètres se baseront notamment sur des statistiques de structures connues d'ARN.

La première partie du document décrira l'état de l'art des connaissances dont nous avons besoin pour notre approche.

Nous développerons dans le chapitre 1 les connaissances de base sur la structure de l'ARN : sa constitution et son repliement. Nous étudierons les approches de repliement existantes, ce qui nous permettra de voir l'émergence des modélisations à gros grains, apportant une nouvelle vision de la molécule. Dans le chapitre 2, nous expliquerons ce que sont la théorie des jeux et la minimisation de regret. Ces outils nous permettront de modéliser le repliement de la molécule et de calculer un équilibre.

La deuxième partie du document se concentre sur un travail préliminaire à notre approche. Nous prouverons dans cette partie la pertinence de notre approche.

Dans le chapitre 3, nous étudierons le repliement d'un objet (une simple chaîne) dans des espaces en deux et en trois dimensions. Nos résultats montreront qu'il est possible de replier une chaîne en trois dimensions pour créer des interactions au sein de cette chaîne tout en trouvant un état stable. Le chapitre 4 décrira une première approche de repliement. Nous décrirons une modélisation de la molécule à gros grain, une étude de plusieurs

paramètres et la conception d'un jeu. Cette première approche, limitée aux molécules de taille moyenne, est fondée sur une discrétisation de l'espace 3D. Les premiers résultats de cette approche montreront que notre méthode permet de trouver des structures au moins aussi proches des structures réelles que les approches actuelles. Le chapitre 5 étudiera notre jeu préliminaire. Nous remarquerons que l'utilisation d'algorithmes de minimisation de regret est plus appropriée que d'autres algorithmes (comme Monte-Carlo). Nous vérifierons aussi la possibilité d'étendre ce travail préliminaire à toutes les molécules, en analysant les points d'améliorations possibles.

La troisième et dernière partie décrira notre méthode suite aux travaux préliminaires.

Dans le chapitre 6, nous modifierons l'approche du chapitre 4 pour l'étendre à toutes les molécules, notamment en s'écartant de la discrétisation de l'espace. Ces changements amélioreront les résultats de notre première approche, tout en nous permettant d'en apprendre plus sur les possibilités du repliement à gros grains. Avec cette méthode, nous arriverons à atteindre notre objectif d'échantillonner l'espace des structures possibles d'un ARN tout en atteignant des structures similaires à celles qui sont recherchées. Le chapitre 7 proposera de trier nos structures pour choisir la meilleure de notre échantillonnage. Nous testerons plusieurs critères de tri pour choisir le plus judicieux. Nous pourrons alors proposer deux structures, dont l'une faisant partie des meilleures de notre échantillonnage.

Partie I
État de l'art

1 La molécule d'ARN

L'ARN est une molécule qui se présente le plus souvent sous la forme monocaténaire (un enchaînement simple de nucléotides formant un brin) dans la cellule. Ce brin se replie sur lui-même en adoptant plusieurs conformations spatiales possibles.

Dans ce chapitre, nous décrirons la structure et le repliement de l'ARN. Dans la section 1.1, nous aborderons la structure globale de l'ARN : sa structure en simple brin, ses interactions et ses repliements possibles. La section 1.2 présentera la structure secondaire qui se forme grâce à des interactions fortes. Nous expliquerons aussi les méthodes de prédiction de cette structure. Nous présenterons dans la section 1.3 sa structure tridimensionnelle et les approches de prédiction de cette structure. Nous verrons que les approches actuelles utilisent différents niveaux de représentation et méthodes de repliement.

1.1 Structure de l'ARN

La plupart des ARN naturels sont présents sous forme *monocaténaire* (simple *brin*) dans la cellule, contrairement à l'ADN qui est sous forme d'un double brin apparié. Ce simple brin est constitué d'un enchaînement de *nucléotides*.

Un *nucléotide* est composé (voir la figure 1.1) d'un groupement phosphate, d'un sucre (ribose dans le cas de l'ARN) et d'une base azotée. Cette base peut être l'Adénine (A), la Guanine (G), la Cytosine (C) ou l'Uracile (U) (voir la figure 1.2) (dans le cas de l'ADN, l'Uracile est remplacé par la Thymine). Le brin d'ARN peut être vu comme une suite de lettres correspondant aux bases azotées de la molécule. Cette représentation est la séquence de l'ARN.

Les brins d'ARN se replient le plus souvent sur eux-mêmes, formant une structure intramoléculaire qui peut être très compacte. La base de cette structure est la formation d'appariements internes entre les bases complémentaires grâce à des liaisons hydrogène.

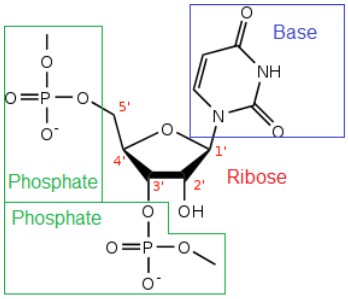


FIGURE 1.1 – **Composition chimique d'un nucléotide.** Un nucléotide est composé d'un ribose, d'une base (ici l'uracile), et d'un groupement phosphate. Les nucléotides sont reliés les uns aux autres par des groupements phosphates.

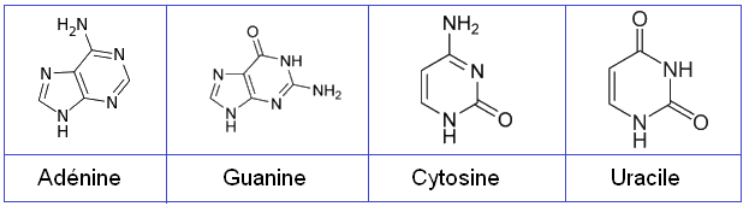


FIGURE 1.2 – **Bases azotées présentes dans l'ARN.** Il existe 4 bases azotées pour l'ARN : l'adénine, la guanine, la cytosine et l'uracile. Ces bases se lieront entre elles pour former des interactions au sein d'une molécule.

Ces appariements sont des liaisons Watson-Crick (l'adénine forme une liaison avec l'uracile, et la guanine avec la cytosine (voir la figure 1.3)) ou des liaisons plus faibles dites *Wobble* (guanine avec uracile) [Crick, 1966]. La description des appariements internes entre les bases d'un ARN s'appelle la structure secondaire. Cette structure secondaire peut être complétée par des interactions à distance, dites tertiaires, qui définissent alors une structure tertiaire qui induira une structure tridimensionnelle de l'ARN.

Le repliement de l'ARN est considéré comme hiérarchique, la structure secondaire (correspondant aux liaisons Watson-Crick) se formant avant les liaisons tertiaires [Tinoco et Bustamante, 1999]. Expérimentalement, les formations de la structure secondaire et de la structure tertiaire peuvent être séparées en modifiant la concentration en *Mg2+* dans la solution d'ARN [Tinoco et Bustamante, 1999, Dima *et al.*, 2005, Li, 2013]. La molécule d'ARN va prendre différentes formes avant une stabilisation de celle-ci.

Le repliement la structure d'une molécule d'ARN peut donc se décomposer en trois

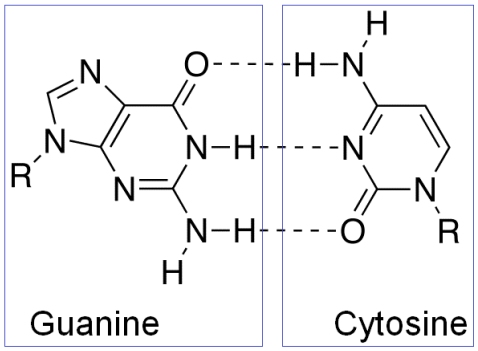


FIGURE 1.3 – **Liaison Watson-Crick.** Une base guanine peut créer trois liaisons hydrogène (lignes pointillées) avec une base cytosine. Ces liaisons vont créer une liaison Watson-Crick.

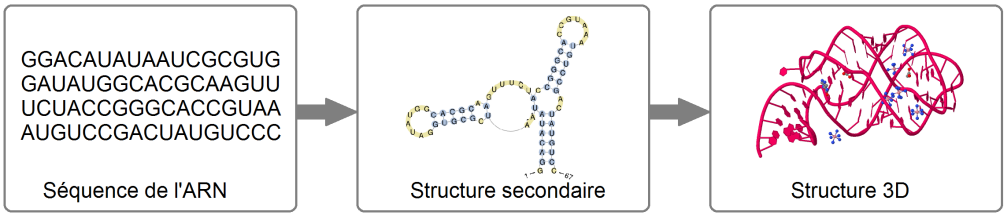


FIGURE 1.4 – **Phases de repliement de l'ARN.** De la séquence d'ARN, les interactions Watson-Crick et *Wobble* créent une structure secondaire qui se repliera en structure 3D grâce à des interactions tertiaires.

phases successives, comme décrit dans la figure 1.4 :
- la séquence, représentant les nucléotides de l'ARN ;
- la structure secondaire, représentant les interactions canoniques Watson-Crick et *Wobble* entre les nucléotides ;
- la structure tridimensionnelle.

L'ARN peut avoir plusieurs configurations stables. Ces configurations dépendent de plusieurs types d'interactions. Elles dépendent des interactions Watson-Crick, qui apportent une contribution électrostatique liée à la formation des liaisons hydrogène entre les bases. Elles dépendent aussi de l'empilement des plateaux des paires de bases (*stacking*) qui donne naissance à des interactions de Van der Waals (des interactions électriques de faible intensité entre atomes). La nomenclature *Leontis-Westhof* [Leontis et Westhof, 2001] classifie les différentes interactions entre les nucléotides en fonction des faces de la base entrantes en jeu (Watson-Crick, *Hoogsteen* ou Sucre) et de l'orientation des nucléo-

tides (*Cis* ou *Trans*). Chaque appariement possible possède une mesure d'*encombrement* (volume pris dans l'espace) et une énergie caractérisant sa solidité. Ces liaisons peuvent se former et se défaire jusqu'à arriver à une forme stable, qui n'est pas obligatoirement unique.

Plusieurs configurations tridimensionnelles stables peuvent alors coexister et demandent beaucoup d'énergie interne et de temps pour passer de l'une à l'autre [Thirumalai et Hyeon, 2009]. L'environnement a aussi un impact important sur la structure finale de la molécule : l'augmentation de la température rend les liaisons chimiques moins stables et la concentration en ions du solvant influe sur la stabilité des liaisons [Tinoco et Bustamante, 1999, Thirumalai et Hyeon, 2009]. Il a été observé [Mandal et Breaker, 2004] que les riboswitches (structures d'ARN servant de régulateur de la traduction) changent de forme en fonction de la présence d'un *ligand* (une petite molécule), changeant ainsi leurs fonctions catalytiques.

Il n'existe donc pas une structure unique pour une molécule d'ARN, mais un ensemble de structures possibles qui se construisent de façon hiérarchique. Il existe de nombreuses méthodes *in silico* pour prédire la structure 3D [Laing et Schlick, 2010], qui utilisent seulement la séquence, ou encore la structure secondaire. Dans le premier cas, plus difficile mais plus idéal, seule la connaissance de la séquence d'ARN suffit au repliement. Avec la structure secondaire, nous avons plus d'informations sur la structure attendue. Les méthodes peuvent aussi utiliser la connaissance des structures d'autres molécules homologues pour prédire une structure. Cela facilite la prédiction, mais demande plus de connaissances sur les molécules d'ARN.

1.2 La structure secondaire

La structure secondaire de l'ARN décrit l'ensemble des appariements internes (Watson-Crick) au sein d'une molécule simple brin [Doty *et al.*, 1959]. Ces appariements Watson-Crick créent une topologie particulière, formée de régions en *hélices* (régions appariées) et de *jonctions* (régions non appariées) (voir la figure 1.5). Chacune de ces régions est un élément de la structure secondaire (*ESS*).

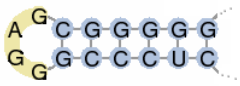


FIGURE 1.5 – **Repliement en hélice et jonction.** Les appariements Watson-Crick et *Wobble* créent des hélices (en bleu) et, par conséquent, des régions non appariées (en jaune), ici une jonction terminale.

Ces différentes régions vont constituer une structure secondaire (voir la figure 1.6) avec plusieurs types de jonctions :
– des *jonctions terminales*, à l'extrémité d'une hélice (ou 1-jonctions) ;
– des *jonctions internes*, qui connectent deux hélices (2-jonctions) ;
– des *jonctions multiples*, qui connectent trois hélices ou plus (n-jonctions avec n hélices).

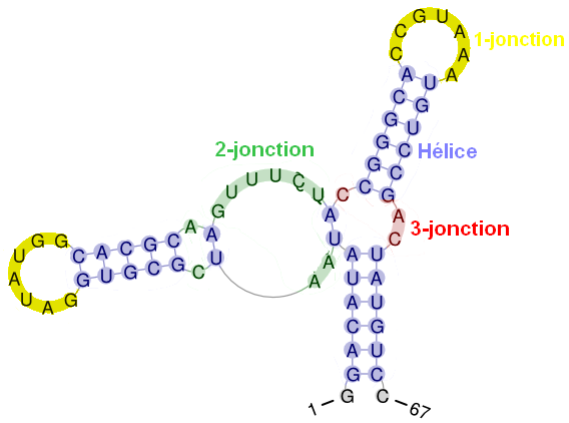


FIGURE 1.6 – **Schéma d'une structure secondaire.** La structure se compose d'hélices (en bleu) et de jonctions (en jaune, vert ou rouge) de différentes tailles. La jonction en rouge est une 3-jonction : une région non appariée reliant trois hélices entre elles. Les jonctions en jaune sont des jonctions terminales (ou 1-jonctions). Les jonctions en vert sont des jonctions internes (ou 2-jonctions).

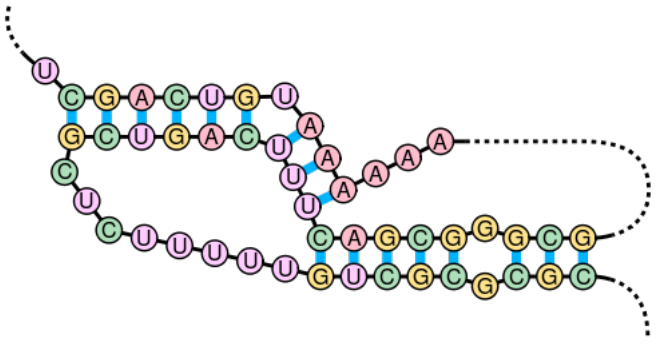


FIGURE 1.7 – **Structure secondaire avec un pseudonœud.** Une zone de la molécule s'est reliée avec une autre zone créant ainsi un pseudonœud. Les brins pointillés correspondent à la suite du brin d'ARN, qui peut se replier de n'importe quelle manière.

En plus des jonctions et des hélices, d'autres cas se présentent comme les *pseudonœuds*.

Les *pseudonœuds* consistent en un brin (reliant deux hélices ou une jonction terminale) créant des liaisons Watson-Crick avec une autre partie de la molécule (voir la figure 1.7).

Les pseudonœuds créent une structure secondaire non arborescente, c'est-à-dire pouvant contenir des cycles, compliquant la prédiction de la structure. La plupart des algorithmes travaillent sans les pseudonœuds, malgré l'importance biologique de ce type de repliement. En effet, une forme avec pseudonœuds possédera des brins s'entremêlant pouvant induire une fonction biologique précise [Staple et Butcher, 2005]. Dans nos travaux, nous utiliserons des structures secondaires sans pseudonœuds, les méthodes actuelles de prédiction de structures secondaires avec pseudonœuds n'étant pas très efficaces.

1.2.1 Prédiction de la structure secondaire

La prédiction de structure secondaire consiste à trouver la ou les conformations possédant une énergie minimale. L'énergie dépend des appariements trouvés. La première utilisation de la programmation dynamique par Nussinov [Nussinov *et al.*, 1978] cherchait à maximiser le nombre total d'interactions Watson-Crick. Les méthodes actuelles, bien plus réalistes, utilisent un modèle d'énergie dit « du plus proche voisin », avec des paramètres déterminés expérimentalement : une énergie est associée à chaque couple de deux paires de bases (GG/CC, AU/UA, etc.) à laquelle nous ajoutons les différentes jonctions. Cette énergie utilise des données thermodynamiques sur les couples créés. Par programmation dynamique, l'algorithme cherche alors la structure optimale minimisant la somme des énergies.

Basées sur cette méthode, plusieurs approches sont disponibles, comme Mfold [Zuker et Stiegler, 1981] ou RNAfold [Hofacker *et al.*, 1994]. Comme une séquence peut se replier en diverses structures secondaires, il est aussi possible de trouver des structures sous-optimales [Zuker *et al.*, 1989]. Il existe aujourd'hui des bases de données de structures secondaires, comme RNA FRABASE [Popenda *et al.*, 2008, Popenda *et al.*, 2010], qui possèdent la séquence et la structure secondaire de plus d'un millier de molécules d'ARN, dérivées des informations déposées dans la base de données *Protein Data Bank*.

1.3 Structure tridimensionnelle

La structure tridimensionnelle prend en compte les appariements autres que canoniques (autres que Watson-Crick et *Wooble*), et les interactions additionnelles à longue distance.

Les appariements non canoniques impliquent toujours des liaisons hydrogène entre les bases. Westhof et Leontis [Leontis et Westhof, 2001] proposent douze grandes familles d'appariements classifiées d'après la face des bases impliquées.

Ces interactions tertiaires sont en nombre non négligeables. En effet, 20 % des nucléotides

forment au moins une liaison non canonique, et environ 21 % ne forment aucune liaison (canonique ou non canonique) [Nasalean *et al.*, 2009].

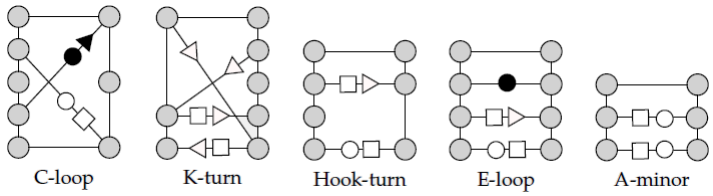


FIGURE 1.8 – **Motifs de 2-jonctions.** Les 2-jonctions peuvent être triées d'après plusieurs motifs. Les ronds gris indiquent les nucléotides, les lignes horizontales ou diagonales indiquent les interactions entre les nucléotides au sein de la 2-jonction. Les interactions sont identifiées suivant la nomenclature Leontis-Westhof [Leontis et Westhof, 2001].

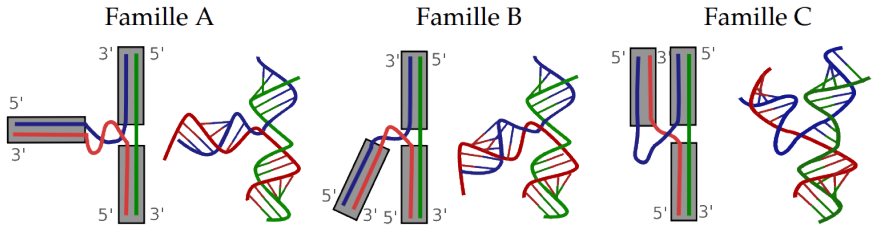


FIGURE 1.9 – **Les trois familles topologiques des 3-jonctions.** Chaque famille (A, B et C) représente une topologie spécifique dans l'espace. Ces familles comportent deux hélices empilées (alignées) par le brin vert et une troisième hélice qui différenciera les familles.

Ces interactions tertiaires induisent des formes tridimensionnelles différentes pour chaque type de jonctions. Nous pouvons alors classifier en familles de forme 3D ces jonctions pour pouvoir les prédire, notamment grâce aux informations de séquence et de structure secondaire locale [Schudoma *et al.*, 2011]. Les 2-jonctions peuvent se replier d'après plusieurs motifs géométriques (K-Turn, Hook-Turn, A-minor, C-Loop, etc.) (voir la figure 1.8) en fonction des structures tertiaires [Lescoute *et al.*, 2005, Djelloul et Denise, 2008]. Les 3-jonctions peuvent se classer en trois familles topologiques [Lescoute et Westhof, 2006] influencées par les informations locales (voir la figure 1.9). Une classification simple de ces familles peut s'effectuer en se basant sur la structure secondaire [Lamiable *et al.*, 2012].

Il existe aussi une classification des 4-jonctions [Laing et Schlick, 2009] (voir la figure

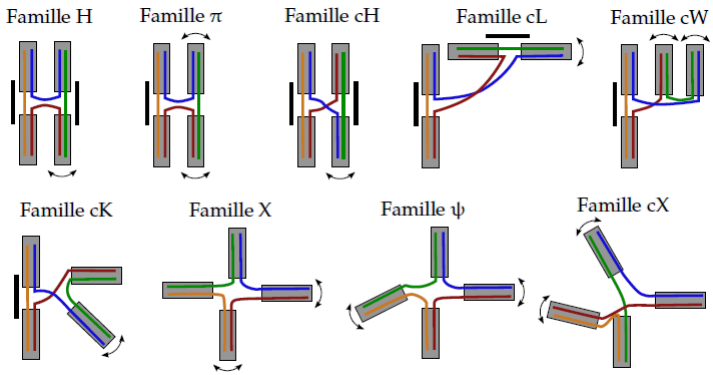


FIGURE 1.10 – **Les neuf familles topologiques des 4-jonctions.** Les neuf familles topologiques pour les 4-jonctions ont été définies par [Laing et Schlick, 2009]. Les traits noirs indiquent un empilement entre deux hélices, et les flèches indiquent une flexibilité de l'hélice.

1.10), qui utilise l'empilement des hélices. Dans ce cas, deux hélices vont s'aligner selon le même axe dans l'espace [Tyagi et Mathews, 2007]. Pour les jonctions plus grandes, il existe moins d'informations permettant une classification claire. Il est néanmoins possible de s'en approcher [Laing *et al.*, 2009, Laing *et al.*, 2012].

1.3.1 Prédiction de la structure 3D

La prédiction de la structure 3D reste une tâche plus complexe que la prédiction de la structure secondaire. Certaines méthodes, comme RNA2D3D [Martinez *et al.*, 2008] ou Assemble [Jossinet *et al.*, 2010], reconstituent la forme connue des hélices puis tentent d'assembler ces dernières dans l'espace via une interface graphique. Il faut alors un expert pour pouvoir reconstituer la molécule. Les approches automatisées utilisent plusieurs informations : la séquence, la structure secondaire et éventuellement les interactions tertiaires. Chaque méthode construit sa représentation de l'ARN, au niveau des nucléotides (voir la figure 1.11) ou à gros grain (voir la figure 1.12). Pour le repliement, ces méthodes peuvent utiliser des potentiels (des fonctions paramétrées pour représenter des connaissances physiques, biologies ou statistiques) ou des bases de données de fragments ou de motifs d'ARN.

La prédiction d'une structure 3D peut s'effectuer grâce à :
- *la dynamique moléculaire*, permettant de simuler l'évolution temporelle (dynamique) d'un système moléculaire ;
- *des fragments et des motifs*, en récupérant des fragments et motifs d'ARN connus pour reconstituer une nouvelle structure 3D ;

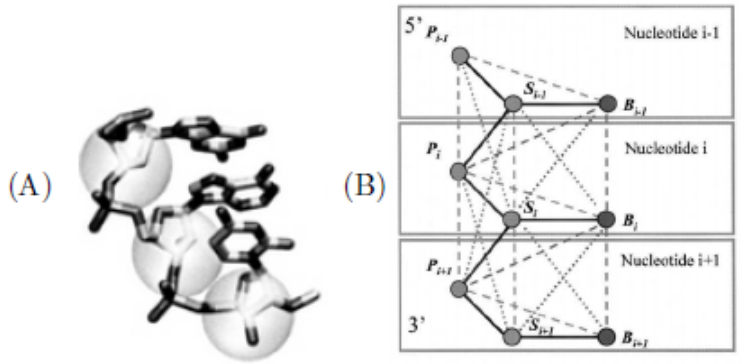


FIGURE 1.11 – **Représentations au niveau nucléotide de l'ARN.** (A) La représentation utilisée par NAST [Jonikas *et al.*, 2009] représente les nucléotides par un point (centré sur l'atome C3').
(B) iFoldRNA [Ding *et al.*, 2008, Sharma *et al.*, 2008] représente chaque nucléotide par trois sphères : une pour le sucre, une pour le phosphate et une pour la base.

– *des potentiels à gros grain*, en représentant l'ARN à gros grain et en utilisant des fonctions de potentiels.

Prédiction par dynamique moléculaire. Une première approche pour le repliement est la simulation de dynamique moléculaire, comme utilisée dans iFoldRNA [Ding *et al.*, 2008, Sharma *et al.*, 2008], NAST [Jonikas *et al.*, 2009] ou SimRNA [Boniecki *et al.*, 2015]. iFoldRNA représente le groupement phosphate, le sucre et la base de chaque nucléotide par trois sphères, puis calcule un potentiel d'après une fonction d'énergie sur les interactions possibles. Une simulation à différentes températures permettra alors d'avoir plusieurs solutions à énergies différentes. NAST représente tout le nucléotide par une seule sphère puis calcule un potentiel d'après des structures connues. NAST replie sa structure grâce à des potentiels utilisés dans le cadre de la dynamique moléculaire. SimRNA représente tout le nucléotide par cinq points (dont trois pour la base) et calcule la structure d'après un potentiel statistique. Ce potentiel dépend des de la géométrie local et des interactions entre les bases. La recherche des conformations se fait ensuite via une méthode de Monte-Carlo. Alors qu'iFoldRNA et SimRNA utilisent seulement la séquence, NAST a besoin de la structure secondaire et des interactions tertiaires pour un bon repliement.

Prédiction par fragments et motifs. Il est aussi possible d'utiliser des bases de fragments ou de motifs. FARNA [Das et Baker, 2007] est une extension de la méthode

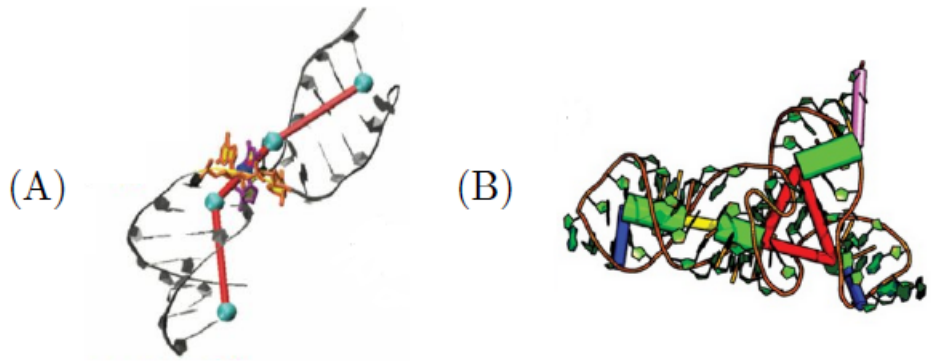


FIGURE 1.12 – **Représentations à gros grain de l'ARN.** (A) RNAJAG [Laing *et al.*, 2013, Kim *et al.*, 2014] représente les éléments de structure secondaire (*ESS*) par un ou plusieurs nœuds d'un graphe.
(B) ERNWIN [Kerpedjiev *et al.*, 2015] représente les éléments de structure secondaire par des cylindres de tailles et de diamètres différents.

Rosetta pour la prédiction de la structure des protéines. Elle modélise le nucléotide par une sphère au centre de la base, puis reconstitue la forme en utilisant une base de données de fragments 3D de trois nucléotides. FARNA tient compte des torsions, collisions et potentiels d'appariement pour optimiser la forme globale grâce à une méthode de Monte-Carlo. MC-Fold/MC-Sym [Parisien et Major, 2008] utilise, lui aussi, une base de données, contenant des motifs 3D de couples de paires de bases. MC-Fold permet de prédire la structure secondaire qui est fournie en entrée pour MC-Sym, qui construit alors un modèle atomique dans lequel sont insérés les motifs de bases par méthode de Monte-Carlo. RNACOMPOSER [Popenda *et al.*, 2012] construit un modèle 3D au niveau atomique, en utilisant la structure secondaire et le dictionnaire de RNA FRABASE [Popenda *et al.*, 2010]. Il récupère de la base de données des fragments de structure secondaire et l'élément 3D correspondant, puis il entreprend une première reconstruction en superposant chaque élément trouvé avant d'affiner le résultat. Cette méthode utilise une grande base de données pour améliorer chaque élément. Ces différentes méthodes travaillent au niveau du nucléotide ou de l'atome.

Prédiction par potentiel à gros grain. Plusieurs méthodes récentes utilisent un niveau de représentation plus haut que le nucléotide, travaillant directement sur un ensemble de nucléotides. RNAJAG [Laing *et al.*, 2013, Kim *et al.*, 2014] et ERNWIN [Kerpedjiev *et al.*, 2015] travaillent sur une représentation à très gros grain de la structure secondaire pour le repliement dans l'espace. RNAJAG utilise une représentation en graphe pour représenter les arrangements spatiaux des hélices autour des jonctions. Des potentiels statistiques sont ensuite utilisés, basés sur des mesures de torsion des jonctions internes

et des rayons de giration. ERNWIN représente les hélices comme des cylindres et travaille sur un potentiel basé sur les forces physiques et sur des potentiels issus de structures connues. Ces deux approches utilisent Monte-Carlo pour arriver au repliement attendu. Une méthode utilisant la théorie des jeux a été étudiée [Lamiable *et al.*, 2013], utilisant aussi un modèle en graphe et une représentation des interactions via des ressorts.

Méthode	Informations utilisées	Représentation	Méthode
iFoldRNA	Séquence	Trois sphères pour un nucléotide	Potentiel d'interactions
NAST	Séquence, structure secondaire, interactions tertiaires	Une sphère par nucléotide	Potentiels statistiques
SimRNA	Séquence	Cinq sphères par nucléotide	Potentiels statistiques
FARNA	Séquence et structure secondaire	Une sphère par nucléotide	Base de données de fragments
MC-Sym	Séquence et structure secondaire	Atomique	Base de données de motif de base
RNACOMPOSER	Séquence et structure secondaire	Atomique	Base de données de motifs et de fragments
RNAJAG	Séquence et structure secondaire	Représentation en graphe de la structure secondaire	Potentiels statistiques
ERNWIN	Séquence et structure secondaire	Représentation en cylindres de la structure secondaire	Potentiels physiques et statistiques

TABLE 1.1 – **Résumé des approches actuelles pour le repliement 3D d'ARN.** Chaque approche utilise des informations, des représentations et des méthodes de repliement différentes. iFoldRNA, NAST, FARNA, MC-Sym et RNACOMPOSER travaillent à grain fin, alors que RNAJAG et ERNWIN travaillent à gros grain. iFoldRNA, NAST, RNAJAG et ERNWIN utilisent des potentiels pour replier leurs structures, alors que FARNA, MC-Sym et RNACOMPOSER utilisent des bases de données de fragments et de motifs. Les fragments sont des morceaux connus d'ARN. Les motifs sont des configurations connues dans l'espace. RNACOMPOSER recherche directement les motifs dont il a besoin dans sa bibliothèque RNA FRABASE.

Ces différentes méthodes sont résumées dans la table 1.1.

1.4 Conclusion

Les méthodes actuelles peuvent représenter l'ARN sous différentes formes : nucléotides, fragment de nucléotides, graphe, cylindre, etc. Les représentations à petite échelle ou utilisant les bases de données fonctionnent bien sur de petites molécules En revanche, cela pose alors problème pour les très grosses molécules, qui demandent beaucoup de temps de calcul mais aussi beaucoup plus d'informations sur ces larges molécules. Par exemple, dans la base de données *PDB*, il n'y a que 67 molécules d'ARN (prises seules) possédant plus de 200 nucléotides, dû aux coûts expérimentaux pour résoudre leur structure. Il est pourtant important de pouvoir travailler sur des molécule aussi larges, un certain nombre étant fonctionnelles [Wilusz *et al.*, 2009].

Les approches à très gros grain montrent des capacités intéressantes pour la prédiction de structure 3D, s'extrayant des difficultés de la gestion d'un nombre important de nucléotides sur les grosses molécules pour se concentrer sur une forme globale. L'utilisation de la structure secondaire permet alors une première vision de la structure 3D en donnant des informations très importantes.

Dans l'idée des approches actuelles, nous avons donc décidé de travailler avec la structure secondaire. Cette structure sera représentée en forme de graphe, donc avec une représentation à gros grain. Nous utiliserons aussi des potentiels statistiques pour le repliement.

2 Théorie des jeux et minimisation de regret

La théorie des jeux est née dans la première moitié du XXe siècle, pour formaliser et résoudre des problèmes de nature économique ou stratégique. Actuellement utilisée dans de nombreux domaines comme l'économie ou plus récemment en informatique pour les réseaux, la théorie des jeux a déjà fait ses preuves en biologie, dans le cadre de la génétique évolutive [Smith, 1979, Smith, 1982].

La théorie des jeux permet de modéliser un système en concurrence sous la forme d'un jeu dans lequel le résultat d'une stratégie par l'un des joueurs ne dépend pas seulement de ses propres stratégies, mais aussi de celles qui sont prises par les autres.

Dans ce chapitre, nous présenterons succinctement la théorie des jeux. Dans la section 2.1, nous expliquerons la théorie des jeux et la formalisation d'un jeu. La section 2.2 se focalisera sur une notion clé : l'équilibre de Nash.

La section 2.3 expliquera comment calculer un équilibre grâce à plusieurs méthodes classiques. De plus, nous décrirons aussi le problème du bandit manchot et les algorithmes de minimisation de regret. Ici, nous nous focaliserons sur des algorithmes classiques que nous utiliserons plus tard (UCB et EXP3). La littérature dans ce domaine est foisonnante (par exemple [Sutton et Barto, 1998, Young, 2004, Cesa-Bianchi et Lugosi, 2006]).

Nous utiliserons par la suite ces notions pour le repliement des molécules d'ARN.

2.1 La théorie des jeux

La théorie des jeux permet de comprendre les systèmes qui peuvent converger vers des situations "rationnelles", nommées équilibres de Nash [Nash, 1950]. Cette théorie étudie la caractérisation des équilibres de jeux (par exemple [Rosenthal, 1973]), la qualité des équilibres [Koutsoupias et Papadimitriou, 1999] et les algorithmes qui calculent des équilibres (par exemple [Berenbrink *et al.*, 2007]).

Un jeu peut être décrit de deux façons différentes. La première est de donner la règle du jeu, son déroulement et la façon dont les joueurs interviennent. C'est alors un *jeu sous forme extensive*. Dans un jeu d'échecs, décrire les règles du jeu est simple, mais donner l'ensemble des stratégies possibles est bien plus complexe, nous décrivons alors le jeu sous forme extensive. La seconde façon de décrire un jeu consiste à donner, pour chaque joueur, l'ensemble de ses stratégies, ainsi que les applications qui associent les gains aux stratégies. C'est un *jeu sous forme stratégique*. Dans un jeu simple de « Pierre, Feuille, Ciseaux », il est aisé de décrire les stratégies et les gains reliés à ces stratégies, la forme stratégique est ainsi celle utilisée. Nous allons nous concentrer sur les jeux sous forme stratégique.

2.1.1 Formalisation des jeux

Un *jeu sous forme stratégique* est un jeu comportant un ensemble N de joueurs, une famille d'ensemble de stratégies $(S_i)_{i\in N}$ et une famille de fonctions de gain $(g_i) : \Pi_{i\in N} S_i \to \mathbb{R}$. L'ensemble des joueurs est supposé fini et non vide, l'ensemble des stratégies sera supposé non vide, et dans le cas où cet ensemble serait fini, c'est un *jeu fini*. Le jeu représente alors les interactions entre les joueurs : chaque joueur $i \in N$ choisit une *stratégie pure* $s_i \in S_i$, les choix étant simultanés, et si $Q = (s_i)_{i\in N}$ est le *profil des stratégies choisies*, le joueur i reçoit le *gain* $g_i(Q)$. En suivant les conventions classiques de la théorie des jeux, de façon abusive, nous écrirons Q_{-i} le vecteur de stratégies de tous les joueurs excepté la stratégie du joueur i.

De plus, les joueurs peuvent avoir une stratégie non-déterministe. Une *stratégie mixte* (voir la définition 2.1) $q_i = (q_{i,1}, q_{i,2}, \ldots, q_{i,|S_i|})$ pour le joueur i correspond à un vecteur de probabilité sur les stratégies pures : la stratégie pure ℓ est choisie avec la probabilité $q_{i,\ell} \in [0, 1]$, avec $\sum_{\ell=1}^{|S_i|} q_{i,\ell} = 1$.

DÉFINITION 2.1 – **Définition d'une stratégie mixte**
Une stratégie mixte pour le joueur i est une loi de probabilités sur S_i.

Les jeux simples sous forme stratégique peuvent être représentés par un tableau (voir la table 2.1).

Voici des exemples de jeux qui ont été intensivement étudiés :
– Un jeu est *jeu de potentiel exact* s'il existe une fonction ϕ définie à partir de l'espace des stratégies pures vers $\mathbb{R}$ telle que pour tout joueur i, pour tout profil de stratégies pures Q, les stratégies s_i, et s'_i, la fonction ϕ satisfait aussi la condition suivante $g_i(s_i, Q_{-i}) - g_i(s'_i, Q_{-i}) = \phi(s_i, Q_{-i}) - \phi(s'_i, Q_{-i})$ Ces jeux sont intensivement étudiés car ils modélisent des jeux d'ordonnancement (placement de tâches sur des serveurs) [Koutsoupias et Papadimitriou, 1999, Berenbrink et Schulte, 2007] ou des jeux de

*Joueur*1 \ *Joueur*2	Pierre	Feuille	Ciseaux
Pierre	**0**, **0**	**-1**,**1**	**1**, **-1**
Feuille	**1**, **-1**	**0**, **0**	**-1**,**1**
Ciseaux	**-1**,**1**	**1**, **-1**	**0**, **0**

TABLE 2.1 – **Exemple de jeu « Pierre, Feuille, Ciseaux ».** Chaque joueur peut choisir comme stratégie Pierre, Feuille ou Ciseaux.
Ici, $S_1 = S_2 = \{"Pierre", "Feuille", "Ciseaux"\}$. Dans chaque case, le gain des joueurs est indiqué. Par exemple, si le joueur 1 choisit Feuille et le joueur 2 choisit Pierre, le gain est de (1,-1), le joueur 1 ayant gagné ($g_1 = 1$), faisant perdre le joueur 2 ($g_2 = -1$). On a donc $g_1(Feuille, Pierre) = 1$.

routage [Roughgarden, 2005, Krichene *et al.*, 2015] ;

– Un jeu *à somme nulle* est un jeu où la somme des gains de tous les joueurs est égale à 0 : c'est-à-dire pour tout profil de stratégies pures Q, $\sum_{i \in N} g_i(Q) = 0$. Le jeu « Pierre, Feuille, Ciseaux » est un jeu à somme nulle (voir la table 2.1).

2.2 Les équilibres de Nash

L'équilibre de Nash [Nash, 1950] est une notion centrale de solution pour les jeux stratégiques. Une *déviation profitable* pour un joueur est une situation où il a intérêt à changer de stratégie (voir la définition 2.2). Si aucun joueur n'a de déviation profitable, le jeu se trouve dans un *équilibre de Nash* (voir la définition 2.3).

DÉFINITION 2.2 – **Définition de la déviation profitable**
Étant donné un profil de stratégies Q, un joueur i a une *déviation profitable* s'il existe une stratégie s_i telle que $g_i(s_i, Q_{-i}) > g_i(Q)$, donc si le joueur peut améliorer son gain en changeant lui seul sa stratégie, sans qu'aucun autre joueur ne change de stratégie.

Par conséquent, dans un équilibre de Nash, un joueur seul ne peut pas améliorer sa situation. Connaissant cette information, tous les joueurs savent que le profil Q va être joué, ils ont tout intérêt à le jouer. L'équilibre de Nash représente donc une situation stable, mais pas obligatoirement la meilleure situation pour chacun des joueurs. Si le jeu admet un équilibre avec des stratégies pures, alors c'est un équilibre de Nash pur. Dans le cas où le jeu admettrait un équilibre avec des stratégies mixtes, c'est un équilibre de Nash mixte.

Dans le cas du jeu Pierre-Feuille-Ciseaux, il n'existe pas d'équilibre de Nash pur. Le seul équilibre de Nash mixte est un profil où les deux joueurs jouent la même stratégie mixte au vecteur de probabilités $(1/3, 1/3, 1/3)$.

DÉFINITION 2.3 – **Définition de l'équilibre de Nash**
Un *équilibre de Nash* est un profil de stratégies Q pour lequel il n'existe pas de déviation profitable.

Ainsi, il n'existe pas toujours d'équilibre de Nash pur, et il a été montré [Nash, 1950] que tout jeu fini $G = (N, (S_i)_{i \in N}, (g_i)_{i \in N})$ possède au moins un équilibre de Nash mixte. Ce théorème garantit l'existence. Par contre, la preuve n'est pas constructive et ne dit pas comment ils peuvent se calculer. Calculer un équilibre mixte pour un jeu fini à N joueurs est PPAD-complet (*Polynomial Parity Argument on Directed graphs*) [Papadimitriou, 2007].

Dans ce cas de complexité, le problème reste difficile [Papadimitriou, 1994] et nous n'avons pas d'algorithmes polynomiaux pour calculer un équilibre dans un jeu quelconque. Discuter la complexité du calcul des équilibres de Nash est lié à la pertinence de cette notion : par exemple, nous pouvons nous dire que si les équilibres sont difficiles à calculer, nous pouvons nous attendre à ce que le système ne tende pas nécessairement vers ces états en un temps raisonnable.

De plus, dans certains jeux, il est facile de calculer un équilibre de Nash. Par exemple, un équilibre de Nash mixte peut se calculer en temps polynomial pour les jeux à deux joueurs à somme nulle. En effet, le jeu peut se modéliser par une matrice A ayant $|S_1|$ lignes et $|S_2|$ colonnes. Comme c'est un jeu à somme nulle, le jeu peut être considéré de la façon suivante : le joueur 1 cherche à maximiser un gain et le joueur 2 a pour objectif de minimiser son coût. En considérant, au temps T, que le joueur 1 (resp. 2) joue en fonction de la stratégie mixte q_1 (resp. q_2), son gain (resp. son coût) est le produit $q_1^T A q_2$. Pour le joueur 1, le gain optimal est $g_1^* = max_x min_y x^T A y$ (pour le joueur 2, le coût optimal est $g_2^* = min_y max_x x^T A y$). Par le théorème min-max, il a été prouvé que le profil (x, y) est équilibre de Nash si le gain de 1 (resp. le coût de 2) est optimal, et réciproquement. Ceci implique que calculer un équilibre de Nash dans un tel jeu est polynomial (il suffit de résoudre le système $g_2^* = g_1^*$). Remarquons que cette méthode nécessite de connaître complètement le jeu.

2.3 Calculer un équilibre

Les développements historiques de la théorie des jeux en mathématiques et en économie ne se sont pas focalisés sur la construction de tels équilibres. C'est pour cette raison que l'on distingue maintenant souvent la théorie des jeux de la théorie algorithmique des

jeux. Pour calculer un équilibre de Nash, nous nous intéressons aux outils liés à la théorie algorithmique des jeux.

2.3.1 Algorithme de construction d'équilibres

Nous nous intéressons à la construction d'équilibres où chaque étape le jeu est répétée. Les joueurs n'ont aucune vision globale du jeu : ils ne connaissent pas les stratégies des autres joueurs ni leur fonction de gain. À chaque fois que le jeu est répété, des joueurs choisissent la stratégie qu'ils vont jouer. D'après leur gain, ils réactualisent leur stratégie en fonction d'un algorithme. Les joueurs adaptent souvent leur stratégie en fonction de leur connaissance locale du système par de petits ajustements afin d'améliorer leur propre gain. L'impact de chaque joueur sur le système est faible. Toutefois, comme le nombre d'acteurs est grand, une évolution globale du système peut se produire.

Ici, nous nous focalisons sur un algorithme qui réalise de petits ajustements (stochastiques ou déterministes). Nous supposons ces ajustements complètement répartis puisque tous les joueurs participant aux jeux ont souvent une vue locale du système, sans vision globale du jeu (information incomplète). À chaque instant, les joueurs choisissent les stratégies qu'ils vont réaliser. En fonction de l'état global du système, ils reçoivent une récompense, que l'on peut voir comme un gain. En fonction de leur passé, ils réactualisent alors leur stratégie afin de maximiser leur gain.

Une première dynamique, qui est la plus naturelle, correspond à la dynamique de la meilleure réponse : seuls les joueurs qui peuvent améliorer leur gain peuvent changer de stratégies.

Par exemple, focalisons-nous sur les jeux de potentiel. Si un seul joueur change de stratégie à la fois, cette dynamique converge vers un équilibre de Nash pur. Rappelons que ces jeux possèdent toujours au moins un équilibre de Nash pur. La preuve [Rosenthal, 1973] peut se voir comme une preuve de dynamique de meilleure réponse : les joueurs jouent chacun leur tour et changent de stratégies en améliorant systématiquement leur gain quand ils le peuvent.

Par des propriétés des jeux de potentiel, si un joueur améliore son gain, alors la fonction de potentiel globale décroît aussi. La suite de profils correspond à une suite de meilleures réponses individuelles. Elle est alors une suite décroissante de la fonction de potentiel. Comme cette suite a un nombre fini de valeurs, une telle suite de profils doit être finie et donc doit atteindre un équilibre de Nash pur [Rosenthal, 1973].

Certains algorithmes, comme *Linear Reward Inaction* (LRI) [Sastry *et al.*, 1994] permettent de calculer directement un équilibre de Nash. L'algorithme LRI (voir l'algorithme 2.1) utilise un vecteur de probabilités pour la sélection de la stratégie. À chaque stratégie pure est associée une probabilités de la choisir. Initialement identiques, ces probabilités

ALGORITHME 2.1 – **LRI [Sastry *et al.*, 1994]**

Algorithme

Paramètres : $b \in [0,1]$ un réel
Initialisation : $P_{i,s} = 1/S; \forall i \in 1, ..., S$
pour chaque *itération t*
    Tirer une stratégie s_i aléatoirement en respectant le vecteur de probabilités $P_i(t)$
    Recevoir un gain $gain_s(t) \in [0,1]$.
    Mettre à jour les probabilités des stratégies :
    si $i = s_i(t)$
        $P_{i,s}(t+1) = P_{i,s}(t) + b(gain_k(t))(1 - P_{i,s}(t))$
    sinon
        $P_{i,s}(t+1) = P_{i,s}(t) - b(gain_k(t))P_{i,s}(t)$
    fin si
fin pour

sont mises à jour à chaque tour de jeu en fonction de la récompense et d'un facteur b. À chaque itération t, le joueur i a seulement besoin de connaître son gain et la stratégie $s_i(t)$ jouée. En fonction de ces informations, le joueur réactualise son vecteur de probabilités $P_i(t+1)$.

En fonction de la valeur de b, l'algorithme converge plus ou moins rapidement vers une solution. Ce facteur permet une augmentation de la probabilité des stratégies sélectionnées où les gains les plus élevés ont été observés. Ici, ce paramètre b vise à être un réel positif proche de 0. Il a été prouvé [Sastry *et al.*, 1994] que, pour des jeux en général, la limite pour $b \to 0$ des dynamiques de cet algorithme est une équation différentielle ordinaire (voir l'équation 2.1) dont les points limites stables (s'ils existent) sont des équilibres de Nash lorsque $t \to \infty$.

$$\frac{dP_{i,s}}{dt} = P_{i,s}(gain_k(P_{i,s}, P_{-i}) - gain_k(P)) \tag{2.1}$$

Si l'équation différentielle ordinaire converge, alors nous pouvons espérer que cet algorithme atteigne des équilibres. Par exemple, pour des jeux [Sastry *et al.*, 1994] à utilité unique (tous les joueurs ont le même gain), cet algorithme converge vers un équilibre de Nash pur. À ce jour, aucun temps de convergence n'a été calculé.

2.3.2 Problème de bandit manchot et minimisation de regret

Nous étudions ici le problème du bandit manchot [Robbins, 1985]. Dans ce problème, un joueur fait face à un certain nombre de machines à sous différentes. À chaque étape du jeu, il choisit une machine sur laquelle il va jouer et reçoit un gain. Le joueur peut

donc mesurer son gain reçu, mais il n'accède pas à d'autres informations. Son but est de maximiser son gain total, soit la somme des gains reçus à chaque étape, en prenant en compte l'historique du jeu.

Ici, les gains reçus par le joueur dépendent d'une loi de probabilité fixe et les machines sont indépendantes. L'objectif du joueur est mesuré en terme de perte par rapport à l'utilisation de la machine rapportant le plus à chaque étape. Le joueur tente alors de minimiser son regret, c'est-à-dire de minimiser la perte qu'il a eue à jouer la mauvaise machine sur certaines étapes du jeu. Il se retrouve face à un dilemme *exploration-exploitation*. En effet, le joueur souhaite accumuler des informations sur les gains reçus sur les machines et jouer la machine qui rapporte le plus. Jouer la machine qui lui semble avoir le gain le plus élevé (*exploitation*) n'est pas une solution, car l'espérance des gains de cette machine peut diminuer avec le temps. Dans le cas inverse, s'il passe trop de temps à tester toutes les machines (*exploration*), le joueur fait diminuer ses gains totaux qu'il aurait pu avoir en utilisant plus souvent la meilleure machine.

Dans le cas du problème de bandit manchot, le joueur cherche à minimiser le regret cumulé. Les algorithmes de minimisation du regret tentent d'optimiser le rapport entre exploration et exploitation.

2.3.3 Les algorithmes de minimisation

Nous présentons ici deux des algorithmes les plus utilisés en minimisation de regret : UCB [Auer *et al.*, 2002a] et EXP3 [Auer *et al.*, 2002b].

Ces algorithmes tentent de minimiser le regret d'un joueur i face à S machines, qui représentent chacune une stratégie, dont la distribution des gains est inconnue, à valeurs dans [0, 1]. À chaque étape, le joueur choisit une machine (une stratégie) s parmi les S machines (stratégies). Chaque machine $s \in S$ a une espérance μ_s, correspondant au gain moyen que le joueur peut recevoir de cette machine. Le jeu continue durant un temps T.

Les gains reçus des machines sont à valeur dans $[0, 1]$. Nous allons étudier deux cas :

1. les machines ont un gain défini par une distribution ;
2. le gain des machines n'est pas défini par une distribution.

UCB

Nous étudions ici un algorithme dans le cas où les machines ont un gain défini par une distribution.

L'algorithme UCB (*Upper Confidence Bound*) [Auer *et al.*, 2002a] (voir l'algorithme 2.2) choisit à chaque tour une stratégie s d'après la moyenne des gains cumulés de cette

ALGORITHME 2.2 – **UCB [Auer *et al.*, 2002a]**

Algorithme

Entrée : S le nombre de stratégies, T temps de jeu total, T_s nombre de fois que la stratégie s a été jouée
Sortie : stratégie choisie
Initialisation : Jouer chaque stratégie s et fixer $\bar{x_s}$
tant que t < T
    pour chaque *stratégie s*
        $indice_s = \bar{x_s} + \sqrt{\frac{2lnt}{T_s}}$
    fin pour
    Choisir la stratégie s qui maximise $indice_s$
fin tant que

stratégie ($\bar{x_s}$) qui représente le terme d'exploitation, et d'après un terme d'exploration.

Nous allons introduire le regret cumulé dans la définition 2.4. Il a été prouvé [Auer *et al.*, 2002a] que le regret cumulé est borné avec l'algorithme UCB (voir le théorème 2.1).

DÉFINITION 2.4 – **Définition du regret cumulé pour un algorithme**
Le regret R pour un algorithme après T étapes est :

$$Regret_T = \mu^* * T - \Sigma_{s=1}^{S} \mu_{s(t)}$$

où μ^* est l'espérance moyenne de la meilleure machine, $\mu_{s(t)}$ l'espérance de la machine s au temps t.

THÉORÈME 2.1 – **Théorème UCB**
Pour tout $S > 1$, si UCB est exécuté sur S machines ayant des distributions de gains arbitraires $g_1, ...g_S$ à support dans [0,1], alors l'espérance du regret cumulé moyen après T étapes de jeu est au plus

$$E(Regret_T) \leq (8\ \Sigma_{s:\Delta_s>0} \frac{logT}{\Delta_s}) + S * \frac{\pi^2}{3}$$

où $\mu_1, ...\mu_s$ sont les espérances de $g_1, ...g_S$ et $\Delta_s = \mu^* - \mu_s$ est la marge entre la machine optimale et la machine k.

L'algorithme UCB est entièrement déterministe. Dans le cas d'une fonction de gain non stochastique (par exemple les gains attribués par un "adversaire"), l'adversaire peut prévoir ses futurs choix et attribuer les gains pour que l'algorithme prenne de mauvaises décisions. Pour gérer les cas des gains non-stochastiques, nous introduisons un nouvel algorithme, EXP3.

EXP3

Nous travaillons ici dans le cas où les machines n'ont pas de gain défini par une distribution. Dans ce cas, les gains sont choisis arbitrairement par un adversaire.

Lorsque le joueur joue contre un adversaire, le modèle de jeu est :

1. l'adversaire distribue les gains avec sa propre politique d'affectation (d'après une distribution des gains décidée au début du jeu) pour chaque machine (supposés à valeurs dans $[0, 1]$, sans les dévoiler au joueur) ;

2. le joueur choisit une machine ;

3. le joueur observe le gain de la machine choisie.

Ce modèle de jeu est dit à *informations partielles* : le joueur ne peut observer que le gain de la machine choisie, il ne connaît pas les gains des autres machines.

L'algorithme EXP3 (*EXPonential EXPloration-EXPloitation*) [Auer *et al.*, 2002b] (voir l'algorithme 2.3 et le théorème 2.2) tente de minimiser le regret à chaque itération du jeu.

Dans l'algorithme EXP3, le joueur choisit une stratégie d'après une distribution de probabilité $P_1(t), ...P_S(t)$. Cette distribution mélange une distribution uniforme (représentant la partie exploration de l'algorithme) et une distribution qui associe à chaque stratégie une probabilité de poids exponentiel d'après le gain cumulé (représentant la partie exploitation).

ALGORITHME 2.3 – **EXP3 [Auer *et al.*, 2002b]**

Algorithme

Entrée : $\gamma \in [0, 1]$
Sortie : Stratégie choisie
Initialisation : $w_i(1) = 1, i \in \{1, ..., S\}$
pour t = 1 à T
    $P_s(t) = (1 - \gamma)\frac{w_i(t)}{\sum_s w_s(t)} + \frac{\gamma}{S}$
    Jouer s_t aléatoirement en respectant les probabilités $P_1(t), ...P_S(t)$
    Recevoir le gain : $x_{s_t}(t) \in (0, 1)$
    pour s= 1 à S
        $\bar{x_s}(t) = \frac{x_s(t)}{P_s(t)}$ dans le cas $s = i_t$, sinon 0
        $w_s(t + 1) = w_s(t)exp(\frac{\gamma \bar{x_s}(t)}{K})$
    fin pour
fin pour

THÉORÈME 2.2 – **Théorème EXP3**
Pour tout $S > 0$ et pour tout $\gamma \in [0, 1]$:

$$G_{max} - E[G_{EXP3}] \leq (e-1)\gamma G_{max} + \frac{SlnS}{\gamma}$$

où G_{max} est le gain maximum si nous jouons la meilleure machine, G_{EXP3} le gain reçu par la machine choisie par l'algorithme EXP3 et $E[G_{EXP3}]$ l'espérance des gains des machines choisis par l'algorithme EXP3.

Dans EXP3, le poids de chaque stratégie prend en compte le gain et la probabilité de la stratégie, permettant aux stratégies avec des probabilités faibles d'avoir des gains plus proches des autres stratégies. EXP3 est réputé être plus robuste sur des données plus bruitées [Auer *et al.*, 2002b], l'algorithme ne travaillant pas sur des distributions de gains connues.

2.3.4 Minimisation de regret et équilibre de Nash

Dans un *équilibre de Nash corrélé*, un arbitre affecte une loi de probabilité sur les profils de stratégies. Chaque joueur est alors informé de cette loi et l'arbitre demande s'il souhaite changer de stratégies. Si aucun joueur n'a d'intérêt à changer de stratégie, alors la distribution de probabilité est un *équilibre corrélé* (voir la définition 2.5). Tout équilibre de Nash est un équilibre de Nash corrélé. Cette notion a été introduite par [Aumann, 1974].

DÉFINITION 2.5 – **Définition d'un équilibre corrélé.**
Soit un jeu à N joueurs, un ensemble de stratégies S_i et un gain g_i pour le joueur i. Soit une loi de probabilité p sur l'ensemble des profils Q. Soit $p(Q)$ la probabilité du profil Q, noté aussi $p(s_i, Q_{-i})$ pour le joueur i. La loi de probabilité p est un *équilibre corrélé* si, pour tout joueur i et toute stratégie $s_i \in S_i$:

$$\forall s'_i \in S_i, \Sigma_{Q_{-i}} p(s_i, Q_{-i}) g_i(s_i, Q_{-i}) \geq \Sigma_{Q_{-i}} p(s_i, Q_{-i}) g_i(s'_i, Q_{-i})$$

Il est possible d'utiliser la minimisation de regret pour converger vers un équilibre de Nash corrélé [Hart et Mascolell, 2000]. Les algorithmes de bandit manchot [Grigoriadis et Khachiyan, 1995, Auer *et al.*, 2002a] sont utiles pour trouver un équilibre de Nash dans les jeux à somme nulle [Grigoriadis et Khachiyan, 1995, Jean-Yves Audibert and, 2009] notamment EXP3 [Auer *et al.*, 2002a]. Il a été prouvé [Grigoriadis et Khachiyan, 1995] que si les deux joueurs utilisent tous les deux l'algorithme EXP3 pour jouer, alors le système converge vers un équilibre de Nash mixte.

2.4 Conclusion

La théorie des jeux a déjà apporté des solutions en biologie, notamment sur la théorie de l'évolution du comportement des espèces. Notre paradigme futur consistera à voir la structure stable d'une molécule d'ARN comme un équilibre de Nash.

La minimisation de regret reste un moyen de nous approcher d'un équilibre de Nash corrélé. Les algorithmes de minimisation de regret présentés ici seront donc utilisés pour minimiser le regret au sein de molécules d'ARN, dans l'optique d'approcher un équilibre de Nash.

Partie II

Étude du repliement d'ARN par théorie des jeux

3 Jeux de repliement dans l'espace

Dans cette partie, nous construisons des jeux préliminaires pour le repliement d'ARN dans un espace 3D. Ces différents jeux montreront l'intérêt de voir le repliement d'une structure d'ARN comme un jeu. En première approche, nous étudions dans ce chapitre des jeux simples de repliement dans des espaces en deux et en trois dimensions. Ces jeux ont pour objectif de replier une chaîne tout en créant le plus d'interactions possible entre les différents nœuds de la chaîne. Ces interactions auront pour objectif de représenter (de façon extrêmement simplifiée) les interactions tertiaires de l'ARN. Nous étudions ici la possibilité de représenter le repliement d'une molécule par un équilibre de Nash.

Dans la section 3.1, nous replierons une chaîne dans une grille à repère orthonormé en deux dimensions. Nous prouverons qu'il n'existe aucun d'équilibre de Nash pur correspondant à un repliement de cette chaîne. Dans la section 3.2, nous poursuivrons l'analyse pour une chaîne dans une grille en trois dimensions. Nous construirons des équilibres de Nash pur correspondant à un repliement de la chaîne.

3.1 Jeu de repliement en deux dimensions

Dans notre jeu, l'ARN est vu comme une chaîne qui se replie dans l'espace. Cette chaîne, composée de plusieurs nœuds, souhaite maximiser le nombre d'interactions entre ces nœuds (représentant les interactions) tout en minimisant ses propres repliements (pour représenter l'énergie nécessaire à une molécule pour changer de conformation). Les nœuds souhaitent ainsi optimiser leur repliement en créant des interactions. Ce type de modélisation a déjà été utilisé pour des problèmes de repliement des protéines, notamment avec un treillis hydrophobe-polaire [Istrail *et al.*, 2009] Nous allons dans un premier temps chercher des équilibres de Nash purs pour une chaîne se repliant dans un espace en deux dimensions. Un équilibre de Nash pur a été défini dans le chapitre 2 (voir la définition 2.3, page 18).

3.1.1 Formalisation d'un jeu de repliement d'une chaîne dans une grille en deux dimensions

Considérons T une grille dans un espace $\mathbb{Z}^2$. La grille T est un réseau infini dont les sommets sont les éléments de $\mathbb{Z}^2$ et dont les arêtes relient les plus proches voisins entre eux (i.e. distants d'une longueur 1). La grille T est muni d'un repère orthonormé. Nous considérons C une chaîne de dimension finie, de N nœuds, reliés deux à deux par *N-1* arêtes. La chaîne C est construite telle que :
– chaque arête de C a une longueur de *1*, correspondant au pas de la grille ;
– chaque nœud de C est sur un sommet de la grille ;
– les nœuds sont ordonnés suivant l'ordre des nœuds dans la chaîne.
Nous souhaitons que la chaîne forme des interactions entre ses différents nœuds. Une interaction L ne peut se créer que si :
– les deux nœuds ne sont pas voisins dans C ;
– les deux nœuds sont à une distance de *1* dans la grille ;
– les nœuds ne sont pas déjà en interaction avec un autre nœud.

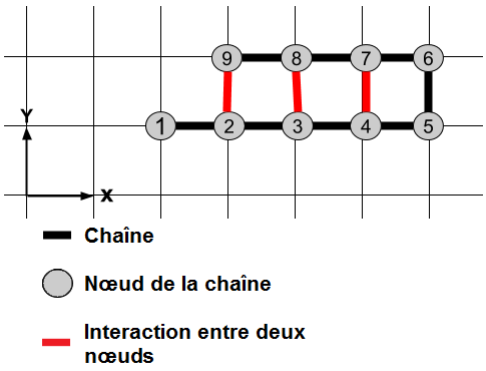


FIGURE 3.1 – **Exemple de repliement d'une chaîne dans un espace à deux dimensions.** Dans une grille carrée à deux dimensions, un chaîne (en noir) et ses différents nœuds (joueurs) se replient d'après les stratégies $S = \{+x, -x, +y, -y\}$. Par exemple, le joueur 2 a pour stratégie $s_2 = +x$, son joueur suivant 3 étant à une position $+x$ (ou (+1,0)) par rapport à lui. Le joueur 5 a pour stratégie $s_5 = +y$, le joueur suivant 6 étant à une position $+y$ (ou (0,+1)) par rapport à lui. Si deux joueurs sont à une distance de 1 dans la grille et ne sont pas voisins dans la chaîne (par exemple les joueurs 4 et 7), ils peuvent créer une interaction (en rouge).

Considérons N l'ensemble des joueurs, correspondant à l'ensemble des nœuds de la chaîne C. Chaque nœud de la chaîne sera un joueur. Chaque joueur aura pour stratégie de placer sur la grille le joueur suivant. Soit S l'ensemble des stratégies possibles pour chaque joueur, avec $S = \{+x, -x, +y, -y\}$. La stratégie $+x$ (resp. $+y$) signifie que le joueur suivant est à la position $(+1, 0)$ (resp. $(0, +1)$) par rapport à la position du joueur actuel.

Le dernier joueur ne joue pas (n'ayant aucun joueur suivant à placer). Soit g_i le gain du joueur i. La fonction de gain est du type : $g_i = g_s + g_L$:

– Le gain g_s correspond au choix de la stratégie, avec pour valeur 0 si la stratégie est la même que le joueur précédent, -1 sinon. Ce gain modélise le fait que la chaîne veut minimiser son nombre de repliement ;
– Le gain g_L correspond à la possibilité de créer une interaction à un autre nœud, avec $g_L = 2$ si une interaction est créée. Ce gain modélise le fait qu'un nœud préfère interagir avec un autre nœud que de ne pas se plier.

Nous n'autorisons pas les joueurs à se superposer, pour éviter que deux parties d'une molécule d'ARN se superposent. Cette interdiction peut être vue comme un gain $g_s = -\infty$ pour le joueur créant la superposition. Nous pouvons alors travailler sur un jeu J avec $J = (N, S, g)$. Le jeu est un jeu séquentiel : les joueurs jouent les uns après les autres. Une fois que l'un de joueurs a choisi une stratégie, le joueur suivant, dans l'ordre de la chaîne, choisit à son tour une stratégie. Une fois que tous les joueurs ont choisi une stratégie, la chaîne sera alors dans une certaine configuration puis nous calculons les interactions possibles pour maximiser le nombre d'interactions présentes. Une configuration possible de repliement est présentée dans la figure 3.1.

Le jeu ainsi créé replie une chaîne souhaitant maximiser son nombre d'interactions tout en évitant des repliements. En effet, si un joueur i se replie (i.e. il choisit une stratégie différente du joueur précédent) sans former lui-même une interaction, son gain sera $g_i = -1$ (comme pour les joueurs 5 et 6 de la figure 3.1). Si le joueur i ne se replie pas et ne forme pas d'interaction, son gain sera de $g_i = 0$. Si le joueur i se replie et forme une interaction, alors son gain sera $g_i = 1$. Enfin, s'il ne se replie pas et forme une interaction, son gain sera de $g_i = 2$ (comme pour les joueurs 4 et 7 de la figure 3.1).

3.1.2 Équilibres de Nash purs pour le jeu de repliement en deux dimensions

Nous avons décrit un jeu simple du repliement d'une chaîne dans une grille 2D. Nous cherchons maintenant s'il existe un équilibre de Nash pur où la chaîne est dans une configuration repliée. Une *configuration repliée* consistera en une configuration où au moins un joueur aura choisi une stratégie différente du joueur précédent.

Nous allons chercher les équilibres de Nash purs pour une chaîne dans un espace en deux dimensions (théorème 3.1).

THÉORÈME 3.1 – **Équilibre de la chaîne en 2D**
Soit T une grille dans un espace $\mathbb{Z}^2$. Soit C une chaîne de N nœuds. Pour le jeu $J = (N, S, g)$, il existe un unique équilibre de Nash pur pour C.

Démonstration. Tout d'abord, considérons la configuration telle que tous les joueurs jouent la même stratégie, correspondant à une ligne droite. Nous appellerons cette configuration la *chaîne droite*. Soit i un joueur de la chaîne. Nous allons considérer deux cas (voir Figure 3.2) :

1. Si i reste sur le même axe, alors son gain est de 0. Dans ce cas, aucun joueur ne peut créer d'interaction ;
2. Si i change de stratégie, alors son gain est de -1. Le joueur i ne peut créer d'interaction avec d'autres joueurs, il ne peut donc pas augmenter son gain. Par conséquent, il n'a pas d'intérêt à changer seul de stratégie, et il choisira la même stratégie que les autres joueurs.

La chaîne droite est donc un équilibre de Nash. Maintenant, nous allons rechercher s'il existe d'autres équilibres de Nash purs.

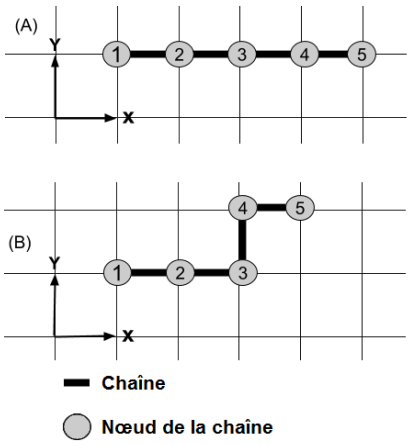


FIGURE 3.2 – **Repliements possibles pour un joueur de la chaîne en deux dimensions.** La courbe noire représente la chaîne C. Dans le cas (A), tous les joueurs jouent la même stratégie (ici la stratégie $+x$). Il n'est pas possible dans ce cas de créer des interactions. Dans le cas (B), seul le joueur 3 a changé de stratégie (la stratégie $+y$), il est néanmoins toujours impossible de créer des interactions. En effet, aucun joueur n'est à une distance de 1 d'un autre joueur non-adjacent.

Supposons que C possède d'autres équilibres de Nash.

Un joueur i est entouré de 4 sommets de grille adjacents, donc 4 joueurs maximum possibles pour créer une interaction. Les joueurs $i-1$ et $i+1$, adjacents à i sont placés sur deux de ces sommets. Il reste deux sommets voisins à i pouvant posséder des joueurs pour une interaction.

Soit Na_C l'ensemble des joueurs de la chaîne C dont l'angle entre les deux arêtes est de

90° ou *-90°*. Un joueur appartenant à Na_C n'a pas la même stratégie que son joueur précédent. Donc son gain g_s lié à la stratégie est $g_s = -1$. Si le joueur ne possède pas d'interaction, il n'aura eu aucun intérêt à avoir changé de stratégie par rapport au joueur précédent et choisira donc la même stratégie que le joueur précédent pour avoir un gain nul.

Nous allons considérer trois cas :

1. Si Na_C possède *0* élément alors C est dans la configuration de la chaîne droite et C est dans un équilibre de Nash ;
2. Si Na_C possède *1* élément alors la distance minimale entre deux joueurs non adjacents est de $\sqrt{2}$ (voir le cas (B) de la figure 3.2). Donc aucun joueur ne peut créer d'interaction, et, par conséquent, le joueur dans Na_C peut changer de stratégies. Les joueurs choisiront de tous jouer la même stratégie ;
3. Si Na_C possède *2* éléments ou plus alors nous sommes dans l'un des cas suivants :
   – Si aucun joueur ne peut créer d'interaction, alors les joueurs $i \in Na_C$ préféreront changer de stratégie pour avoir un gain nul. Cette configuration est un équilibre de Nash pur ;

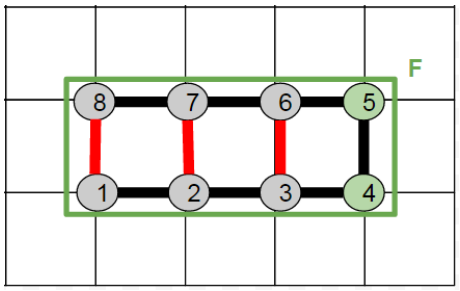


FIGURE 3.3 – **Repliement possible d'une chaîne en deux dimensions avec sa frontière.** La courbe noire représente la chaîne C. Chaque rond représente un joueur. La frontière F (en vert) contient tous les joueurs et toutes les interactions. Le joueur 4 et le joueur 5 (en vert) appartiennent à Na_C mais ne peuvent pas créer de lien. Ces joueurs peuvent changer de stratégie pour améliorer leur gain.

   – Si au minimum deux joueurs ont pu créer une interaction, alors il existe un cycle correspondant à l'interaction et à la chaîne reliant les deux joueurs de cette interaction. Cela signifie que le cycle va former un polygone dans cet espace et une frontière F. Dans un espace en deux dimensions, la frontière F délimite une partie *intérieure* et un partie *extérieure*. La partie *intérieure* contient tous les nœuds de F et tous les nœuds contenus dans le polygone délimité par F. Soit i un joueur de Na_C appartenant à F. S'il existe un joueur v voisin de i (non adjacent à i dans la chaîne C) et à l'extérieur de F, nous considérons une nouvelle frontière F contenant la précédente et les autres joueurs en fonction

de l'interaction entre i et v. Nous réitérons ce processus jusqu'à ce qu'il existe un joueur $j \in Na_C \cap F$ ne possédant pas de joueur sur ses sommets voisins à l'extérieur de F. Dans ce cas, le joueur j ne peut pas créer d'interaction avec un autre joueur (voir Figure 3.3). Il n'a pas d'intérêt à garder sa stratégie et choisira la stratégie de son joueur précédent. S'il existe un interaction, il existera toujours un joueur $j \in Na_C \cap F$ ne pouvant créer d'interaction, ce joueur changera donc de stratégie.

Pour une chaîne C dans T, il n'existe donc qu'un seul équilibre de Nash pur (la configuration de la chaîne droite). □

3.2 Jeu de repliement en trois dimensions

En deux dimensions, un jeu simple de repliement ne permet pas de trouver un équilibre de Nash pur où la chaîne se retrouve repliée pour créer des interactions.

Notre optique est de replier une molécule d'ARN dans un ensemble en trois dimensions. Nous allons rechercher les équilibres de Nash correspondant à des chaînes repliées en 3D. Pour ce jeu, nous considérerons un jeu semblable à celui en deux dimensions et nous le transposerons en trois dimensions.

3.2.1 Formalisation dans une grille en trois dimensions

Soit T une grille dans un espace $\mathbb{Z}^3$. La grille T est un réseau infini dont les sommets sont les éléments de $\mathbb{Z}^3$ et dont les arêtes relient les plus proches voisins entre eux (i.e. distants d'une longueur 1). La grille T est muni d'un repère orthonormé. Nous considérons C une chaîne de dimension finie, de N nœuds, reliés deux à deux par *N-1* arêtes. La chaîne C est construite telle que :
– chaque arête de C a une longueur de *1*, correspondant au pas de la grille ;
– chaque nœud de C est sur un sommet de la grille ;
– les nœuds sont ordonnés suivant l'ordre des nœuds dans la chaîne.

La chaîne souhaite former des interactions entre ses différents nœuds. Une interaction L ne peut se créer que si :
– les deux nœuds ne sont pas voisins dans C ;
– les deux nœuds sont à une distance de *1* dans la grille ;
– les nœuds ne sont pas déjà en interaction avec un autre joueur.

Pour construire un jeu, considérons N l'ensemble des joueurs, correspondant à l'ensemble des nœuds de la chaîne C. Chaque nœud de la chaîne sera un joueur. Chaque joueur aura pour stratégie de placer sur la grille le joueur suivant. Soit S l'ensemble des stratégies possibles pour chaque joueur, avec $S = \{+x, -x, +y, -y, +z, -z\}$. Le dernier joueur ne joue pas (n'ayant aucun joueur suivant à placer). Soit g_i le gain du joueur i. La fonction

de gain est du type : $g_i = g_s + g_L$.
- Le gain g_s correspond au choix de la stratégie, avec pour valeur 0 si la stratégie est la même que le joueur précédent, -1 sinon. Ce gain modélise le fait que la chaîne veut minimiser son nombre de repliement. ;
- Le gain g_L correspond à la possibilité de créer une interaction à un autre nœud, avec $g_L = 2$ si une interaction est créée. Ce gain modélise le fait qu'un nœud préfère interagir avec un autre nœud que de ne pas se plier.

Nous n'autorisons pas les joueurs à se superposer, deux parties d'une molécule d'ARN ne pouvant se superposer. Nous pouvons alors travailler sur un jeu J avec $J = (N, S, g)$. Ce jeu est un jeu séquentiel : les joueurs jouent les uns après les autres. Une fois que l'un de joueurs a choisi une stratégie, le joueur suivant, dans l'ordre de la chaîne, choisit à son tour une stratégie. Une fois que tous les joueurs ont choisi une stratégie, la chaîne sera alors dans une certaine configuration, nous calculons les interactions possibles pour maximiser le nombre d'interactions présentes.

3.2.2 Équilibre de Nash pur en 3D

Nous allons démontrer le théorème 3.2, permettant de trouver un équilibre de Nash pur pour la chaîne dans un espace en trois dimensions.

THÉORÈME 3.2 – **Équilibre de la chaîne *droite* en 3D**
Soit T une grille dans un espace $\mathbb{Z}^3$. Soit C une chaîne élémentaire de N nœuds. Si tous les joueurs jouent la même stratégie, alors la configuration de la chaîne C est un équilibre de Nash pur. Cette configuration sera nommée la configuration de la *chaîne droite*.

Démonstration. Si tous les joueurs jouent la même stratégie, alors leur gain g_s est de 0. Les joueurs jouant la même stratégie, tous les joueurs sont sur le même axe. Par conséquent, aucun joueur n'est à une distance de 1 avec un autre joueur non-adjacent. Aucun joueur ne peut créer d'interaction et tous les joueurs ont pour gain $g = 0$.

Soit i un joueur de C. Si i change de stratégie, alors le gain lié à la stratégie est de g_{s_i} = *-1*. Soit V_- l'ensemble des joueurs précédents du joueur i dans la chaîne C et V_+ l'ensemble des joueurs suivants du joueur i dans la chaîne C. Les joueurs de V_- restent à la même distance du joueur i, donc i ne peut créer d'interaction avec les joueurs appartenant à V_-. Les joueurs appartenant à V_+ gardent la même stratégie que les joueurs V_-. Le joueur i est à une distance de *1* avec le joueur $i + 1$ mais ne peut créer d'interaction avec lui car ces joueurs sont adjacents dans la chaîne C. Le joueur i est à une distance de $\sqrt{2}$ avec le joueur $i + 2$, il ne peut donc pas créer d'interaction avec ce joueur, ainsi qu'avec les autres joueurs de V_+ qui seront à une distance supérieure à $\sqrt{2}$. Le joueur i n'a pas d'intérêt à changer de stratégie et choisira la même stratégie que les joueurs V_-. Si les

joueurs jouent la même stratégie, alors nous nous retrouvons dans le cas de la chaîne *droite*, ce qui équivaut à un équilibre de Nash pur. □

Nous avons donc un équilibre de Nash pur, mais qui ne correspond pas à une configuration repliée de la chaîne. Nous allons démontrer que d'autres configurations sont aussi des équilibres de Nash dans notre jeu J en 3D (théorème 3.3).

THÉORÈME 3.3 – **Existance d'un équilibre de Nash pour la chaîne repliée en 3D**
Soit T une grille dans un espace $\mathbb{Z}^3$. Soit C une chaîne élémentaire de N nœuds, $N \geq 8$. Il existe un équilibre de Nash pur où la configuration de la chaîne C n'est pas la chaîne droite.

Remarquons, par étude de cas ci-dessous, que pour une chaîne avec $N < 8$ nœuds, il existe un seul équilibre de Nash pur correspondant à la configuration de la chaîne droite.

Démonstration. Considérons les cas suivants :

1. Dans le cas où $N < 4$, les joueurs sont sur un plan et la distance minimale entre deux joueurs non-adjacents est $\sqrt{2}$. Donc il n'y a pas d'interaction et les joueurs n'ont pas d'intérêt à jouer des stratégies différentes ;
2. Dans le cas où $N = 4$, $N = 5$, $N = 6$ ou $N = 7$, il existe des configurations où deux joueurs peuvent créer une interaction. Pour $N = 4$, la configuration pour créer une interaction est la configuration formant un carré entre les 4 joueurs. Dans ce cas, deux joueurs peuvent former une interaction, mais deux autres se retrouvent avec un gain $g = -1$ et changeront de stratégie pour revenir à la chaîne droite. Pour créer une configuration avec une ou plusieurs interactions, plusieurs joueurs doivent choisir une stratégie différente des joueurs précédents pour créer une configuration repliée. La configuration repliée permettant de créer des interactions est une configuration proche d'une forme cubique. Dans ces configurations, il existe toujours un joueur i ayant une stratégie différente de son joueur précédent (donc $g_{s_i} = -1$) mais ne pouvant créer d'interaction (donc $g_i = -1$). Par conséquent, le joueur changera de stratégie pour améliorer son gain. La configuration permettant de créer des interactions n'est pas stable et les joueurs choisiront tous la même stratégie. La chaîne reviendra alors à la configuration de la chaîne droite.
3. Dans le cas où $N = 8$, il existe une conformation où tous les joueurs ne jouent pas la même stratégie que leur joueur précédent pour créer une forme cubique. Dans ce cas, chaque joueur a pour gain $g = -1 + 2 = 1$ (voir la figure 3.4). Ici, si un joueur change de stratégie, alors son gain peut passer à $g = 0 + 2$, mais dans ce cas-là il annulera une interaction. Le joueur dont l'interaction a été brisée changera de

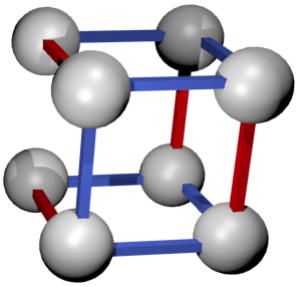


FIGURE 3.4 – **Équilibre de Nash dans la grille 3D.** La chaîne possède ici $N = 8$ joueurs, chacun représenté par une sphère. La chaîne est représentée par la ligne bleue. Chaque joueur est en interaction (les lignes doublées rouges) avec un autre joueur. Ainsi, chaque joueur à un gain de 2 plutôt que de 0 dans le cas de la configuration de la chaîne *droite*.

stratégie, jusqu'à briser toutes les interactions. Nous arriverons alors à la chaîne *droite*. La forme *cubique* est donc un équilibre de Nash pur ;

4. Dans le cas où $N > 8$, nous pouvons utiliser l'équilibre du cas $N = 8$. Dans cette configuration, tous les joueurs ont créé des interactions et ne peuvent plus en créer une autre. Si des joueurs sont ajoutés à cette configuration alors ils peuvent choisir de prendre la même stratégie que le dernier joueur. Dans cette configuration, les joueurs de la configuration où $N = 8$ ne changent pas de stratégies et les joueurs ajoutés forment une chaîne *droite*. Pour $N > 8$, il existe un équilibre de Nash où la configuration n'est pas la configuration *droite*.

Par conséquent, si nous étudions une chaîne dans l'espace, il est possible de trouver un équilibre de Nash pur qui crée des interactions entre les nœuds et forme une chaîne repliée.

□

Nous avons pu prouver qu'il existe des équilibres de Nash purs correspondant à la conformation d'une chaîne repliée. Notre étude a été réalisée sur une grille munie d'un repère orthonormé. Cette grille permet à chaque sommet d'avoir 6 sommets voisins (en trois dimensions). Sur une grille permettant à chaque sommet d'avoir plus de voisins, les possibilités de repliement seraient plus nombreuses.

3.3 Conclusion

Dans ce chapitre, nous avons recherché des équilibres de Nash purs correspondant à une chaîne créant des interactions en minimisant ses efforts de repliement. Dans un espace en deux dimensions, la chaîne ne peut pas former d'interactions et être dans un équilibre de Nash pur en même temps. Dans un espace en trois dimensions, la chaîne peut se replier pour créer des interactions tout en étant dans un équilibre de Nash pur.

Cette étude nous confirme l'intérêt de notre paradigme pour la suite de notre étude. Une chaîne longue peut représenter une longue molécule qui se replie. Avoir trouvé un équilibre de Nash pur correspondant à un repliement de la chaîne, donc d'une possible molécule, nous permet de supposer que nous pouvons replier des molécules en cherchant cet équilibre.

Nous allons considérer que le repliement d'un ARN peut être vu comme un équilibre de Nash. Dans le chapitre suivant, nous utiliserons alors des algorithmes de minimisation de regret pour approcher ces équilibres.

4 Modélisation du repliement

Dans le chapitre précédent, nous avons proposé un modèle simple de jeu pouvant se replier pour former des interactions. Nous avons aussi vu, dans l'état de l'art, qu'il est possible de représenter l'ARN comme un graphe à gros grain.

Dans ce chapitre, nous effectuons un travail préliminaire de repliement de l'ARN à très gros grain par un modèle de jeu.

Ce travail s'intéresse aux molécules de taille moyenne (environ 50/100 nucléotides) et dont la taille maximale des jonctions est de 3 hélices adjacentes (3-jonctions). Cette méthode permettra de calculer un échantillonnage des structures possibles, dont certaines seront proches de la structure réelle.

Nous étudierons dans la section 4.1 la représentation des joueurs. Chaque élément de structure secondaire sera représenté comme un ou plusieurs nœuds d'un graphe, chaque nœud représentant un joueur. La section 4.2 expliquera l'ensemble de stratégies des joueurs. Ces stratégies représentent des repliements dans un espace 3D discret. Le gain des joueurs sera détaillé dans la section 4.3, avec la notion de potentiel entre les différents joueurs. Avec ces informations, nous modéliserons un jeu qui nous permettra de calculer un équilibre dans la section 4.4. La section 4.5 présentera les choix que nous avons faits pour paramétrer le jeu, avec une classification des molécules pour leur repliement. Pour étudier ces résultats, nous expliquerons dans la section 4.6 la méthode d'évaluation. La section 4.7 comparera nos résultats avec les structures recherchées (les molécules réelles). Une comparaison avec les autres structures trouvées par d'autres approches existantes sera détaillée dans la section 4.8.

4.1 Représentation de la molécule

Pour modéliser une molécule d'ARN, nous utilisons des informations sur sa séquence et sa structure secondaire.

Séquence : GGACAUAUAAUCGCGUGGAUAUGGCACGCAAGUUU-CUACCGGGCACCGUAAAUGUCCGACUAUGUCC
Structure secondaire : .(((((((..(.(((((.......))))).).....).((((((.......))))))..)))))).

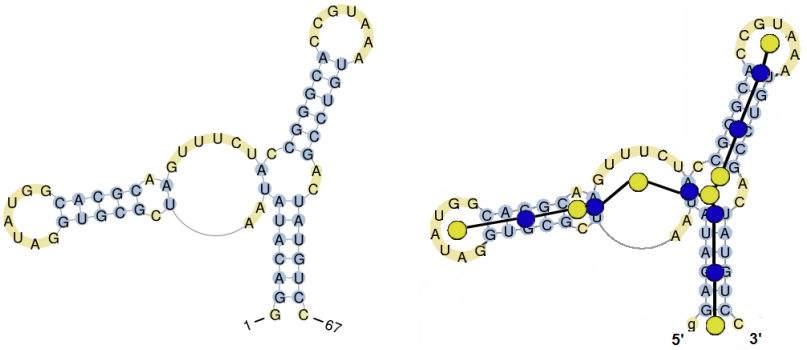


FIGURE 4.1 – **Informations de séquence et de structure secondaire.** La séquence contient la liste des nucléotides de la molécule. Sa structure secondaire est représentée par la méthode *dot-bracket* : les points représentent les nucléotides non appariés, et les parenthèses représentent les liaisons canoniques entre un premier nucléotide (parenthèse ouvrante) et un second (parenthèse fermente correspondante). En dessous, la structure secondaire est représentée avec en bleu les hélices et en jaune les jonctions. Sur la représentation graphique de droite, nous y superposons le graphe que nous utiliserons pour représenter la structure secondaire.

Notre modèle à gros grain représente la structure secondaire comme un graphe, les nœuds décrivant des éléments de structure secondaire (qui seront plus tard des joueurs). Des représentations en graphe ont déjà été développées pour l'ARN [Lamiable *et al.*, 2013, Laing *et al.*, 2013, Kim *et al.*, 2014], chacune prenant en compte différents approches du repliement.

Chaque élément de la structure secondaire (*ESS*) est représenté par un ou plusieurs nœuds dans le graphe (ces éléments ont été présentés dans la section 1.2, page 6) (voir la figure 4.1) :

- 1− **ou 2-jonction**. Les 1− et 2-jonctions sont représentées par un seul nœud ;
- **Hélices**. Les hélices sont représentées par un nœud pour chaque groupe de 5 paires de bases. Une hélice peut faire un tour complet avec 11 paires de bases [Sinden, 2012]. Nous avons voulu représenter un tour complet d'hélice avec plusieurs nœuds. Cela nous permet de mieux gérer le repliement de l'hélice d'après sa longueur ;
- **3-jonction**. Les 3-jonctions sont représentées par deux nœuds (voir la figure 4.2). Le premier nœud représente l'empilement des hélices prédit via une partie de la méthode de classification des familles de [Lamiable *et al.*, 2012]. L'empilement des hélices

s'opère alors sur le plus petit brin non apparié de la 3-jonction (d'après le sens de synthèse des acides nucléiques en cas d'égalité). Pour cela, le brin non apparié le plus court correspondra à l'empilement, obligeant les hélices adjacentes à être alignées. Le nœud correspondant à l'empilement possède trois arêtes, deux des arêtes étant donc automatiquement alignées. Le second nœud représente la position globale de la jonction et la catégorie de la famille [Lamiable *et al.*, 2012].

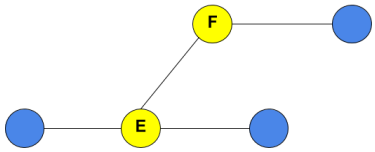


FIGURE 4.2 – **Modélisation des 3-jonctions.** Les nœuds de la jonction (en jaune) sont reliés aux trois hélices adjacentes (en bleu). Le nœud **E** représente l'empilement entre les deux hélices (nœuds bleus alignés), et le nœud **F** représente la famille de la jonction (la direction de la troisième hélice).

Dans ce chapitre, nous ne travaillerons que sur les molécules possédant comme plus large jonction des 3-jonctions pour une simplification de la modélisation. Dans la troisième partie du document, nous travaillerons sur tous les types de jonctions.

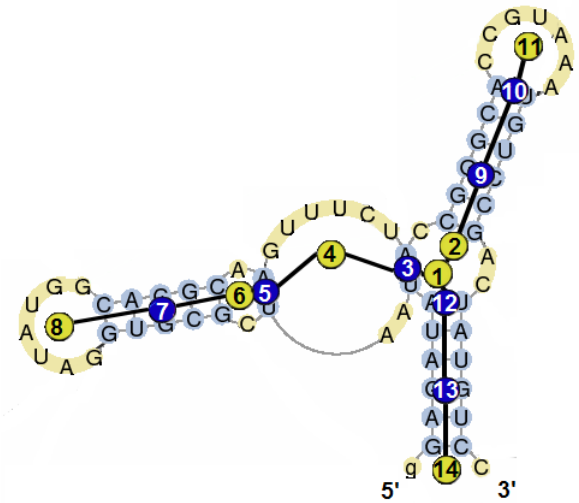


FIGURE 4.3 – **Modélisation en graphe de la molécule 4FE5.** Les nœuds du graphe sont bleus pour les hélices, jaunes pour les jonctions. Les nombres indiquent l'ordre des nœuds. Cet ordre débute sur la jonction de taille la plus élevée (d'après l'ordre 5'-3'), puis suit l'ordre 5'-3'.

En résumé, chaque nœud représente une partie ou tout un élément de la structure secondaire.

Les arêtes du graphe relient entre eux chaque nœud du graphe. Deux éléments adjacents dans la structure secondaire sont reliés par une arête dans le graphe.

Une fois le graphe créé, nous lui donnons un sens de parcours (voir la figure 4.3), correspondant à un parcours en profondeur en partant de la jonction qui a le plus d'hélices. S'il y en a plusieurs, nous prenons la première jonction dans l'ordre 5'-3' (sens de synthèse des acides nucléiques, l'extrémité 5' correspondant à l'extrémité terminée par un phosphate et 3' à l'extrémité terminée par un ribose). Ce parcours permet donc de déterminer un ordre entre les nœuds.

4.1.1 Plongement dans l'espace 3D

Pour permettre la représentation 3D de la molécule grâce à notre graphe, nous plongeons les différents nœuds dans un espace 3D. Chaque nœud est positionné au niveau du barycentre des nucléotides de l'élément qu'il représente (voir la figure 4.4). Le nœud d'une 2-jonction représente ainsi le barycentre des nucléotides non-appariés de la jonction. Pour les nœuds d'une hélice, chacun est placé au niveau du barycentre du groupe de 5 paires de bases ou moins qu'il représente. Dans le cas des 3-jonctions, le nœud représentant la famille de la jonction est au barycentre des nucléotides non-appariés et le nœud de l'empilement est au barycentre entre les paires de bases adjacentes des hélices empilées. Cette représentation utilise les positions connues des atomes de la molécule. Chaque nœud de notre graphe représente la position connue d'un élément, grâce aux informations que nous pouvons récupérer des bases de données, comme la *Protein Data Bank*.

4.1.2 Modélisation sur une grille

Comme dans le chapitre précédent, la projection d'un problème dans une grille permet de limiter les configurations possibles. Nous allons donc positionner nos nœuds dans un espace discret, une grille tridimensionnelle, pour mieux maîtriser les différents repliements. Nous construisons la grille de telle façon que chaque nœud de notre graphe soit placé dans cet espace discret (sur un sommet) tout en représentant au mieux le barycentre des nucléotides de son élément de structure secondaire d'après la position connue (comme précédemment). Il existe plusieurs grilles possibles, nous travaillons ici sur une grille triangulaire [Gillespie *et al.*, 2009] (voir la figure 4.5 pour un exemple en deux dimensions).

Nous travaillons sur cette grille pour sa régularité (tous les points de la grille ont le même nombre de voisins, et toutes les paires de points adjacents sont à la même distance) et pour sa densité élevée (un point de la grille triangulaire possède 12 voisins, contre 6 dans une grille cubique). Ce type de grille est un bon compromis entre une densité élevée (ce

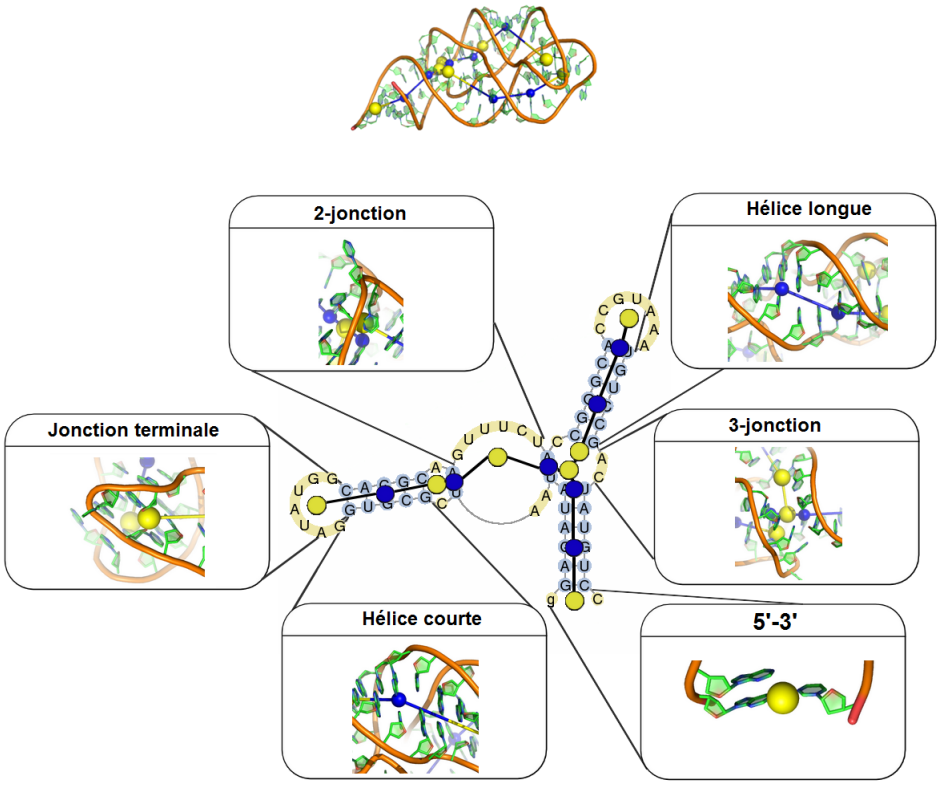


FIGURE 4.4 – **Plongement du graphe de 4FE5 dans un espace 3D.** Les nœuds du graphe sont bleus pour les hélices, jaunes pour les jonctions. Les hélices longues (représentées par un ou deux nœuds) contiennent au moins 6 paires de bases, et les hélices courtes aux plus 5 paires de bases. Les nucléotides non-appariés des extrémités 5' et 3' sont aussi représentés comme une jonction terminale. Nous observons pour chaque type de nœud son plongement dans la structure 3D.

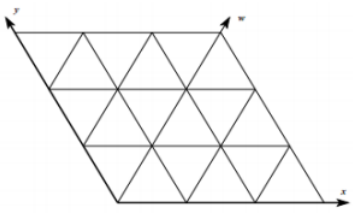
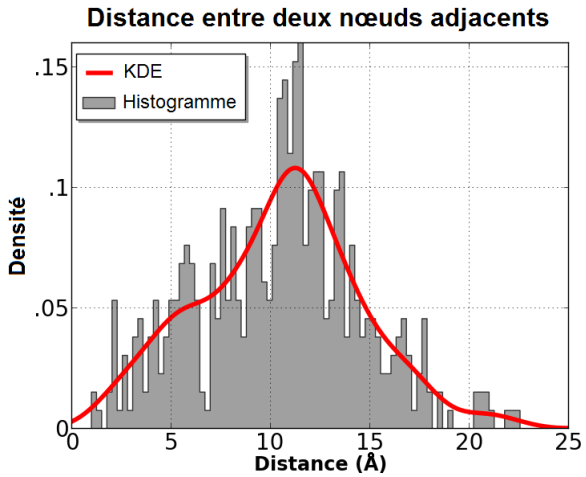


FIGURE 4.5 – **Grille triangulaire en 2D.** La grille triangulaire (ici en deux dimensions) permet à chaque sommet d'avoir 6 voisins en deux dimensions, et 12 voisins dans une grille 3D.

qui permet une certaine souplesse dans les directions possibles) et la possibilité de les dénombrer facilement. Pour construire la grille, nous avons besoin de fixer la longueur d'un pas de grille, de façon que chaque élément de structure secondaire soit placé au mieux sur un point de la grille par rapport à un placement « idéal » dans un espace continu. Nous allons donc calculer statistiquement quel pas de grille nous devons utiliser pour représenter au mieux les molécules réelles. Pour cela, nous utilisons un *ensemble de référence* venant de la base de données de [Bernauer *et al.*, 2011] (disponible dans la table annexe AT.1). Les structures secondaires sont récupérées de la base de données RNA FRABASE [Popenda *et al.*, 2010]. Chaque molécule a été plongée dans notre graphe d'après le calcul des nœuds vu précédemment. Les informations des positions des atomes et des nucléotides proviennent des fichiers PDB de la *Protein Data Bank*. Nous avons ensuite étudié la distance entre les nœuds adjacents dans le graphe (les données sont disponibles dans la figure 4.6)



	Min	1er Qu.	Médian	Moyenne	3ème Qu.	Max
Distance entre deux nœuds	2.068	8.601	11.280	11.080	13.450	26.960
Distance entre deux nœuds d'une 3-jonction	2.068	5.008	5.636	6.683	6.983	18.570

FIGURE 4.6 – **Statistiques sur la distance entre les nœuds adjacents.** L'histogramme représente la répartition de la longueur entre deux nœud adjacents. Nous observons deux pics sur la densité des distances. Le tableau détaille les statistiques que nous avons réalisées sur l'*ensemble de référence*. La première ligne du tableau indique les statistiques de distance entre deux nœuds adjacents quelconques, la deuxième ligne indique la distance pour deux nœuds d'une 3-jonction.

Les mesures montrent une distribution bi-modale, avec des pics à 5.6 Å et 11.2Å. Le pic

à 5.6 Å est plus haut entre deux nœuds d'une même jonction (au sein d'une 3-jonction). Si les nœuds sont d'un type différent, le pic se retrouve près de 11.2 Å. Un pas de grille de 5.6 Å permet de gérer les deux cas.

Ainsi, deux nœuds voisins dans le graphe seront à distance d'un ou deux pas. Les nœuds sont positionnés à 1 pas (à 5.6 Å) l'un de l'autre s'il s'agit de 3-jonctions, de petites hélices ou petites 2-jonctions (inférieures à 6 nucléotides). Dans les autres cas, le pas est de 2 (de 11.2 Å).

Les nœuds pourront se déplacer dans la grille, tout en restant sur les sommets de la grille et en respectant les distances entre chaque nœud que nous avons imposées.

Dans la suite, chaque nœud du graphe sera un joueur, et nous prendrons en compte le type d'élément de structure secondaire que le nœud représente pour notre jeu.

4.2 Stratégies des joueurs

Chaque joueur est maintenant représenté par un nœud dans une grille triangulaire en trois dimensions. Pour pouvoir simuler le repliement, chaque joueur a pour rôle de choisir une direction dans l'espace pour placer le joueur suivant d'après l'ordre du graphe. L'espace est très discrétisé et les mouvements limités pour faciliter la recherche de solutions. L'ensemble de stratégies de chaque joueur est un ensemble de 12 directions dans l'espace (voir la figure 4.7), donné par l'ensemble des directions des arêtes de la grille triangulaire.

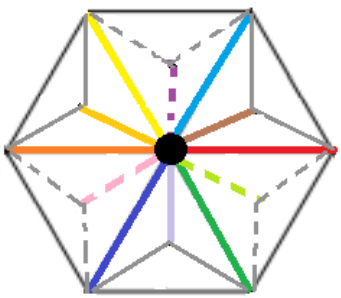


FIGURE 4.7 – **Stratégies sur la grille triangulaire.** Chaque stratégie est une direction sur la grille triangulaire. Chacune des directions est représentée par une couleur différente.

Une fois la stratégie choisie, le joueur utilisera la direction (i.e. la stratégie) pour placer dans la grille un autre joueur, son joueur suivant d'après l'ordre de parcours du graphe (voir la figure 4.8). Prenons par exemple un joueur de type *Hélice* avec comme position

dans l'espace (0,0,0) et comme stratégie la stratégie $+x$ dans l'espace, le joueur suivant étant distant de 2 pas (un joueur 2-jonction par exemple), il aura pour position (2,0,0) dans la grille. La figure 4.9 décrit l'effet de la stratégie d'un joueur sur la structure.

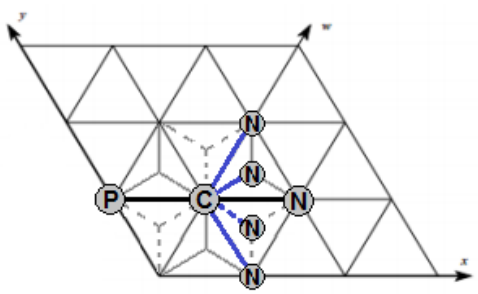


FIGURE 4.8 – **Application de la stratégie d'un joueur.** Le joueur courant (C) choisit une stratégie pour placer dans une direction donnée le joueur suivant (N), en prenant en compte sa position par rapport à son joueur précédent (P). Dans cet exemple, le joueur (C) a la possibilité de positionner le joueur (N) en ne déviant que de 60^o maximum par rapport à la stratégie du joueur précédent.

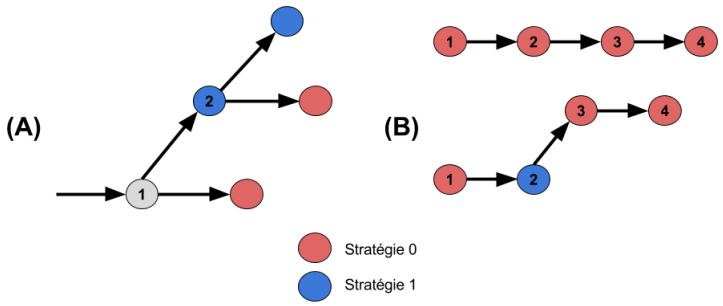


FIGURE 4.9 – **Choix et impact des stratégies.** Les cercles représentent les joueurs, et les flèches représentent les directions vers lesquelles un joueur peut en positionner un autre. Chaque stratégie est représentée par une couleur différente. La figure (A) montre deux stratégies possibles (0 et 1) pour le joueur 1 et le joueur 2. La figure (B) montre le changement de stratégie pour le joueur 2. Dans le premier cas, le joueur 2 a choisi la stratégie 0 (rouge) et suit ainsi la stratégie du joueur précédent. Dans le second cas, le joueur 2 a choisi la stratégie 1 (bleue) et modifie ainsi la structure de la molécule.

À chaque tour de jeu, le joueur a un ensemble de stratégies inclus dans l'ensemble de stratégies des 12 directions, mais prenant en compte des impossibilités et des restrictions. En effet, il faut retirer les directions provoquant une superposition ou un croisement entre

deux joueurs. Nous ajoutons à cela des restrictions d'après le type de joueurs. Chaque élément de structure secondaire ne va pas se replier de la même façon dans l'espace. Nous imposons donc des restrictions d'après le type d'élément de structure secondaire d'un joueur (voir le tableau 4.1 pour un résumé des restrictions).

Élément de structure secondaire	**Restrictions**
Hélice (inférieure à 6 paires de bases)	Pas de divergence (0° maximum)
Hélice (supérieure ou égale 6 paires de bases)	Pas de divergence et 60° maximum sur le deuxième joueur
2-jonctions (plus petit brin non-apparié possède moins de 2 nucléotides)	60° maximum
2-jonctions (plus petit brin non-apparié possède au minimum 2 nucléotides)	Aucune restriction
3-jonctions	Aucune restriction

TABLE 4.1 – **Restrictions de l'ensemble de stratégies pour le jeu préliminaire.** Chaque joueur représente un élément ou une partie de la structure secondaire. D'après la structure qu'il représente, nous imposons des restrictions sur le repliement possible pour ce joueur. Les hélices sont les éléments les moins souples de notre structure. Si une 2-jonction possède un brin non apparié très court, nous lui imposons une restriction. Dans tous les autres cas, le joueur pourra choisir la stratégie qu'il souhaite.

Joueurs de type hélice. Les hélices sont une structure possédant des interactions fortes, ce qui leur impose de ne pas pouvoir se replier complètement. Les hélices restent dans un même axe, même s'il est possible d'observer une légère inflexion si l'hélice est grande [Chastain et Tinoco Jr, 1991]. Si l'hélice est petite (inférieure à 6 paires de bases), elle doit choisir la même direction que le joueur précédent pour éviter un repliement. L'ensemble de stratégies est restreint à la même stratégie que le joueur précédent. Dans le cas où l'hélice posséderait plus de 5 paires de bases, elle est représentée par deux joueurs ou plus. Le premier est soumis aux mêmes restrictions que l'hélice de moins de 6 paires de bases. Les autres joueurs de l'hélice ont pour ensemble de stratégies toutes les stratégies divergeant de 60° par rapport à la stratégie du joueur précédent.

Joueurs de type 2-jonction. De manière générale, l'ensemble de stratégies n'a pas de restriction. Si l'un des brins non-appariés de la 2-jonction possède moins de 2 nucléotides, nous considérons que l'un des brins n'est pas assez souple pour le repliement. Dans ce cas, l'ensemble de stratégies est restreint à toutes les stratégies divergeant de 60° maximum par rapport à la stratégie du joueur précédent.

Joueurs de type 3-jonction. Dans les 3-jonctions, nous avons deux types de joueurs : le joueur d'*empilement* et le joueur de la jonction (voir la figure 4.2). Ces deux joueurs n'ont

pas de limitation dans leur stratégie. Il est à noter que le joueur d'*empilement* placera automatiquement une des hélices empilées par rapport à l'autre, pour pouvoir assurer l'alignement des deux hélices. Cela permet de représenter l'information d'empilement pour les familles [Lamiable *et al.*, 2012] des 3-jonctions.

Il est possible de forcer les hélices à adopter une configuration droite quelle que soit leur longueur, ne leur laissant qu'une stratégie possible (la même direction que le joueur précédent). Nous appellerons hélices *rigides* le cas où les hélices n'ont donc qu'une stratégie possible. Cela permet de modéliser plus fortement la rigidité des hélices. Cette rigidité permet d'avoir des structures moins compactes, la rigidité induisant des molécules plus longiligne.

4.3 Gain des joueurs

Une fois les joueurs repliés dans l'espace, il faut leur attribuer un gain permettant de quantifier la qualité du repliement. Un repliement de qualité est un repliement qui se rapproche des formes réelles des molécules. Pour vérifier si la structure est correcte, nous allons étudier si les distances entre les joueurs non adjacents sont proches de celles qui sont observables sur des structures connues de molécules. Cette distance optimale sera représentée par un potentiel statistique qui servira à calculer le gain des joueurs.

DÉFINITION 4.1 – **Définition du gain d'un joueur i.**
Soit N le nombre de joueurs. Pour tout joueur k, notons $gain_k$ son gain et t_k son type d'élément de structure secondaire. Le gain d'un joueur i est défini comme :

$$gain_i = \sum_{j=1}^{N} Potentiel(t_i, t_j, d_{ij}) \tag{4.1}$$

où *Potentiel* est une fonction qui associe à une couple d'éléments de structure secondaire un nombre réel dépendant de la distance d_{ij} entre les deux joueurs.

Nous cherchons à connaître la distance optimale entre deux types de structures secondaires (nos joueurs) et de tendre vers cette distance. Nous allons calculer un potentiel entre deux joueurs, qui dépend des types d'élément de structure secondaire des joueurs. Ce potentiel correspond à la fonction de distance optimale : plus la distance entre les deux joueurs est proche de la distance optimale, plus le potentiel sera élevé. Pour chaque couple de joueurs, nous calculons leur potentiel. Le gain (voir la définition 4.1) sera la somme de ces potentiels entre le joueur et tous les autres joueurs. Chaque joueur va tenter de maximiser son gain.

Pour chaque couple d'éléments de structure secondaire (hélice/1-jonction, hélice/hélice,

etc..), nous calculons la distance d entre les deux nœuds sur l'*ensemble de référence*, sans la molécule d'identifiant *1Z58*. La prise en compte de la molécule *1Z58*, possédant bien plus de nucléotides que les autres (2880 nucléotides contre 414 nucléotides pour la seconde plus grosse), change de manière significative les résultats des potentiels. De plus, cette molécule n'est pas représentative des molécules que nous étudions actuellement. En effet, la distance optimale sans cette molécule est aux alentours de 20 Å, alors qu'elle passe à 90 Å si nous la prenons en compte. Cette distance étant très élevée, elle est supérieure à la distance maximale observée entre deux joueurs pour de nombreuses molécules observées. Nous ne prendrons donc pas en compte la molécule *1Z58* de l'*ensemble de référence* lors du paramétrage de nos potentiels. Les distances d calculées pour chaque couple nous fournissent la répartition des distances et nous permettent de trouver une distance optimale pour chaque couple. Cette distance optimale sera utilisée pour calculer les fonction des potentiels.

Pour représenter ce potentiel entre deux joueurs, nous allons tester plusieurs potentiels (voir la définition 4.2). Ces potentiels standards sont paramétrés d'après des statistiques de l'*ensemble de référence* (sans la molécule *1Z58*). Ils visent à reproduire la densité des distances entre les joueurs. Le potentiel de Lennard-Jones a été inversé par rapport à son utilisation usuelle (par exemple dans l'étude de la cohésion entre deux atomes) pour nous permettre de maximiser notre gain.

DÉFINITION 4.2 – **Définitions des potentiels.**
Soit i et j deux joueurs, avec respectivement t et t' comme types d'élément de structure secondaire. Soit d la distance entre ces joueurs. Les potentiels possibles entre les joueurs i et j sont :

- **Lennard-Jones**. $-A_{t,t'} \cdot ((\frac{B_{t,t'}}{d})^{12} - (\frac{B_{t,t'}}{d})^{6})$, où $A_{t,t'}$ et $B_{t,t'}$ sont les paramètres de la fonction.
- **Lennard-Jones seuillé**. Si $Lennard\_Jones(t, t', d) < 0$, alors $Lennard\_Jones = 0$.
- **Gauss**. $\Sigma_{i=0}^{m_{t,t'}} C_{i,t,t'} \cdot \frac{1}{\sigma_{i,t,t'}\sqrt{2\pi}} e^{-\frac{(d-\mu_{i,t,t'})^2}{2\sigma^2_{i,t,t'}}}$, avec $m_{t,t'}$ le nombre de fonctions de Gauss utilisées pour le calcul de la fonction. $C_{i,t,t'}, \sigma_{i,t,t'} et \mu_{i,t,t'}$ sont les paramètres de la fonction.

Nous utiliserons alors différents potentiels :

- *Lennard-Jones*, qui ne prend en compte qu'une unique distance optimale, car elle ne contient qu'un seul pic ;
- *Lennard-Jones seuillé*, qui réduit la partie *répulsive* (la partie à la puissance 12), qui diminuait fortement le gain du joueur si la distance entre lui et l'autre joueur est inférieure à un certain seuil ;
- *Gauss*, qui propose plusieurs distances optimales (en sommant plusieurs fonctions de Gauss).

La figure 4.10 permet de voir une représentation de ces potentiels. Dans cet exemple, nous pouvons calculer le potentiel entre un joueur i de type hélice et un joueur j de type 2-jonction. D'après la distance entre ces deux joueurs, nous calculons un potentiel. Nous effectuons la même opération avec les autres joueurs $j \neq i$ pour calculer le gain du joueur i, en sommant ces potentiels.

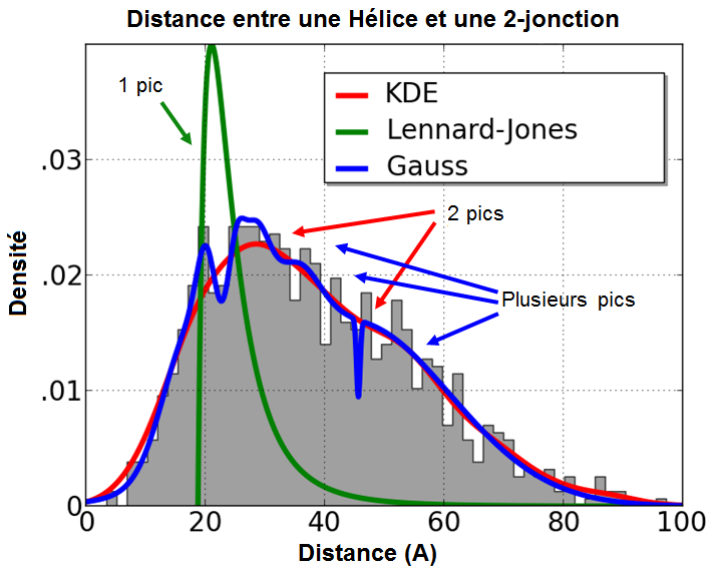


FIGURE 4.10 – **Exemple de potentiels entre deux joueurs.** L'histogramme représente les distances entre un joueur hélice et un joueur 2-jonction pour les molécules de l'*ensemble de référence* (sans *IZ58*). Nous affichons en rouge la densité des distances, en vert le potentiel de Lennard-Jones et en bleu le potentiel de Gauss. Le potentiel de Gauss vise à mimer l'histogramme, alors que le potentiel de Lennard-Jones essaie d'imposer une distance optimale (ici au niveau du premier pic à 20.5 Å).

Les potentiels utilisés (Lennard-Jones, Lennard-Jones seuillé, Gauss) ont des effets sur la conformation d'après le nombre de pics et d'après la partie *répulsion* du potentiel. Les potentiels de Lennard-Jones et de Lennard-Jones seuillé forcent, avec un seul pic, les joueurs dans une conformation spécifique. Cependant, la partie répulsive du potentiel de Lennard-Jones empêche parfois d'arriver à des conformations plus repliées, ce qui permet d'obtenir une meilleure structure pour certaines molécules. Le potentiel de Gauss permet, grâce à plusieurs distances optimales et à l'absence de partie répulsive, d'arriver à une grande variété de conformations.

4.4 Calcul de l'équilibre

Notre jeu possède maintenant ses caractéristiques :

– les joueurs sont des nœuds d'un graphe représentant la structure secondaire ;
– les stratégies sont les directions possibles dans l'espace 3D ;
– le gain dépend de la distance entre les joueurs non adjacents.

Pour le repliement, chaque joueur doit choisir une direction (ce qui va créer une conformation spatiale), puis calculer son gain pour mettre à jour les vecteurs de probabilités (voir la définition 2.1 du chapitre 2, page 16). Le déroulement du jeu peut se voir en plusieurs parties (voir l'algorithme 4.1) :

1. chaque joueur choisit une stratégie et l'applique ;
2. chaque joueur calcule son gain et met à jour son vecteur de probabilité ;
3. l'opération est répétée jusqu'à la fin du jeu.

Le jeu est un jeu séquentiel : les joueurs jouent les uns après les autres, dans l'ordre défini dans la section 4.1. Si, lors d'un tour de jeu, l'un des joueurs n'a pas de stratégie à jouer (par exemple, si les seules stratégies autorisées imposent une superposition de deux joueurs), alors le tour est annulé et une pénalité (le plus petit gain possible pour le joueur) est donnée aux joueurs ayant déjà joué (pour éviter qu'ils rejouent les stratégies ayant menées à cette impossibilité).

ALGORITHME 4.1 – **GARN**

Algorithme

Données : Séquence, Structure Secondaire, T nombre de tours
Sortie : Structure 3D repliée
Création du graphe d'après la structure secondaire
Initialisation du jeu
pour chaque tour de jeu $t < T$
    Un ensemble I de joueurs choisit une stratégie et l'applique
    Un ensemble J de joueurs calcule son gain et met à jour ses vecteurs de probabilités
fin pour
Retourne la structure

Soit N l'ensemble des joueurs.
Dans le déroulement (AA), I=J=N.
Dans le déroulement (OA), $I = \{i\}, i \in N$, $J = N$, pour chaque $i \in N$.
Dans le déroulement (OO), $I = J = \{i\}, i \in N$, pour chaque $i \in N$.
Le jeu se déroule pendant T tours. Ce nombre de tours dépend du nombre de joueurs N, avec $T = 500 * |N|$.

Méthodes de calcul. Sur cette base, nous testons trois méthodes de calcul différentes : (AA), (OA) et (OO), décrites dans l'algorithme 4.1 (la lettre A correspond à *All* et la lettre O à *One*). Dans la séquence (AA), tous les joueurs (les uns après les autres) vont choisir une stratégie puis l'appliquer. Une fois le repliement effectué, tous les joueurs calculent leur gain et mettent à jour leur vecteur de probabilités. Dans la séquence (OA), ce n'est qu'un seul joueur qui choisit une stratégie et l'applique, puis tous les joueurs calculent leur gain et procèdent à la mise à jour de leur vecteur de probabilités. Enfin, dans la séquence (OO), un seul joueur choisit une stratégie, l'applique, et met ensuite à jour son vecteur de probabilités.

Ces méthodes permettent plusieurs types d'exploration de l'espace des solutions : en effet, si un seul joueur change de stratégie, la conformation globale changera moins que si tous les joueurs changent de stratégie. L'importance de la différence des conformations est aussi moindre si un seul joueur met à jour son vecteur de probabilité.

Dans la séquence (AA), avant la mise à jour du vecteur de probabilité, tous les joueurs auront choisi un repliement. Cette méthode donne accès à des conformations très différentes d'un tour de jeu à l'autre. La différence entre deux mesures de configurations consécutives est plus importante, ce qui donne un échantillonnage plus large, donc plus efficace. Dans la séquence (OA), tous les joueurs mettent à jour leur stratégie, mais un seul joueur change de stratégie à chaque tour, ce qui réduit la largeur de l'échantillonnage. La séquence (OO) diminue l'influence des autres joueurs. Lorsqu'un joueur joue, ce même joueur et met à jour son vecteur de probabilité à chaque étape, sans mettre à jour les vecteurs des autres joueurs. L'échantillonnage reste très local. Dans la séquence (AA), la première configuration sera complètement indépendante de la configuration initiale lors du calcul du gain, cette dernière n'a alors aucune importance. Dans les séquences (OA) et (OO), la configuration initiale a un impact sur le gain du premier tour, mais cet impact diminuera fortement lors du deuxième tour. Au début de chaque jeu, la configuration initiale sera calculée aléatoirement.

Algorithmes de minimisation de regret. Sur ces différentes méthodes, nous appliquons deux algorithmes : UCB et EXP3 (voir la section 2.3.3, page 21). La performance des algorithmes utilisés dans le jeu dépend principalement du gain (qui permet de mettre à jour les vecteurs de probabilités). L'algorithme UCB fonctionne mieux lorsque l'échantillonnage et le gain ne comportent pas un large espace de solution. [Auer *et al.*, 2002a] a montré que cet algorithme n'est pas robuste face à des solutions contenant beaucoup de bruit (plusieurs conformations proches mais moins bonnes). L'algorithme EXP3 est plus robuste sur des données bruitées [Auer *et al.*, 2002b], donc sur des structures de molécules plus variées et sur des gains plus complexes. Ces algorithmes nous permettent de tendre vers un regret minimum pour chaque joueur, et donc vers un équilibre de Nash corrélé [Hart et Mascolell, 2000].

4.5 Choix finaux pour GARN

Dans les sections précédentes, nous avons présenté plusieurs paramètres de jeu : différents potentiels, différentes méthodes de calcul, etc. Pour régler les paramètres de notre méthode, nous l'avons exécutée et évaluée sur l'*ensemble de référence* (voir la table en annexe AT.1). Pour cette étude, nous avons analysé les différences entre les paramètres molécule par molécule, pour en sortir un paramétrage global.

Tous les paramétrages n'ont pas le même effet selon les molécules. Pour pouvoir apporter le meilleur ensemble de paramètres possible pour une molécule, nous avons différenciées les molécules d'après deux critères : la présence (ou non) de 3-jonctions et la *flexibilité* de la molécule (indiquée par le rapport entre le nombre d'hélices et de jonctions). Nous avons ainsi dégagé 3 familles de molécules, avec chacune ses paramètres spécifiques :

Famille 1 : Molécules sans 3-jonction. Dans le cas où la molécule n'a pas de 3-jonction, les meilleurs paramètres sont la méthode de calcul (OA) avec l'algorithme UCB et le potentiel de Lennard-Jones seuillé. Les hélices resteront *rigides* dans cet ensemble de paramètres.

Famille 2 : Molécules avec 3-jonctions et un grand nombre d'hélices. Si la molécule contient au moins une 3-jonction, les paramètres sont la méthode de calcul (AA) et l'algorithme EXP3. Si le ratio hélice/jonction est supérieur à 1.5, alors le potentiel sera Lennard-Jones.

Famille 3 : Molécules avec 3-jonctions et un petit nombre d'hélices. Si le ratio hélice/jonction est inférieur à 1.5, nous utiliserons donc le potentiel de Lennard-Jones seuillé. Les autres paramètres seront les mêmes que ceux de la famille 2.

Nos choix sont basés sur les considérations suivantes, tirées de l'examen de chaque molécule de l'*ensemble de référence*. Les 3-jonctions, en modifiant même légèrement leur stratégie, vont influer fortement sur l'ensemble de la structure. Avec notre jeu, si un joueur 3-jonction change sa stratégie, il change alors fortement le gain de tous les joueurs, d'un tour à l'autre. L'algorithme EXP3 est efficace dans ce type d'environnement. En revanche, dans les structures ne contenant pas de 3-jonction, l'algorithme UCB donne de meilleurs résultats : un changement dans la stratégie d'un joueur influencera moins le gain des autres joueurs. Il n'y a que des modifications moins contraignantes de la structure, et l'algorithme UCB permet une optimisation locale. De la même manière, ces molécules se replient mieux dans une séquence (OA), où les modifications subtiles pourront avoir une influence directe sur tous les joueurs. Dans le cas des molécules sans 3-jonction, le repliement s'opère principalement par les 2-jonctions. Pour ne pas brouiller le repliement des 2-jonctions, les hélices sont *rigides* et n'impacteront pas la structure par leur stratégie. Pour le choix du gain, lorsqu'une molécule contient un nombre relativement important de joueurs de type d'élément de structure secondaire hélice (donc un ratio hélice/jonction

important), la partie répulsive du potentiel devient plus déterminante. Comme il y a plus d'hélices dans ce cas, les contacts entre les joueurs doivent être moins proches. La partie répulsive de Lennard-Jones permet d'éviter les rapprochements. Inversement, lorsqu'il y a plus de joueurs de type d'élément de structure secondaire jonctions, il y aura plus d'interactions entre ces joueurs. Dans ce cas-là, les contacts rapprochés sont plus fréquents : la partie répulsive doit être diminuée, comme dans le Lennard-Jones seuillé. Dans tous les cas, les potentiels avec une seule distance optimale (Lennard-Jones et Lennard-Jones seuillé) permettent d'obliger la structure à choisir une conformation repliée d'après cette distance optimale.

4.6 Méthode d'évaluation

Pour savoir si nos résultats sont proches des molécules réelles, nous allons utiliser la méthode d'évaluation classique présentée dans cette section.

Dans un premier temps, nous récupérons dans la *Protein Data Bank* la molécule cristallographiée (la *native*) qui représente la forme que nous souhaitons atteindre. Dans la suite de ces travaux, nous nommerons la structure de la molécule provenant de la *Protein Data Bank* la structure *PDB*. Si nous nous comparons avec une autre approche, nous récupérons les structures de sortie de ces approches. Ces structures de molécules sont ensuite projetées dans notre modèle à gros grain pour être comparées à nos résultats. Pour les logiciels qui utilisent des modèles à gros grain, nous utiliserons leurs résultats avec leur modèle, ne pouvant plonger leur modélisation dans la nôtre.

DÉFINITION 4.3 – **Définition de la RMSD**
Soit une molécule M. Soit deux conformations A et B de cette molécule. Soit N le nombre de joueurs dans la représentation de la molécule M. Soit i un joueur de cette représentation, A_i la position spatiale de ce joueur dans la conformation A et B_i sa position spatiale dans la conformation B. La RMSD entre les deux conformations est :

$$RMSD(A,B) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||A_i - B_i||^2}$$

Pour mesurer la similarité entre deux molécules, nous calculons la déviation de la racine de la moyenne des carrés, *Root Mean Square Deviation* [Kabsch, 1976] (notée RMSD) (voir la définition 4.3) . Cette méthode est classiquement utilisée dans la comparaison de structures 3D de molécules [Maiorov et Crippen, 1994]. Elle consiste à calculer la distance spatiale entre deux points des molécules. En premier lieu, les deux structures que nous comparons sont superposées pour être les plus proches possible. Ensuite, la distance moyenne entre les différents points des molécules (dans notre cas, les joueurs)

est calculée.

Une RMSD très faible signifie que la superposition des deux molécules est très bonne, la distance spatiale entre des joueurs représentant un même élément étant faible. Inversement, une RMSD très forte signifie une structure très éloignée. Il est difficile de trouver la limite entre une bonne et une mauvaise RMSD. Il est donc important de comparer des valeurs de RMSD qui concernent des molécules de tailles similaires. Notons aussi que de nombreuses petites dissimilarités peuvent entraîner une RMSD élevée alors que les structures sont globalement similaires.

4.7 Solutions de GARN

Nous avons appliqué et évalué notre approche sur toutes les molécules de l'*ensemble de test* (voir la table AT.2) dans l'optique de les comparer avec les structures PDB. Les structures secondaires sont récupérées de la base de données RNA FRABASE [Popenda *et al.*, 2010]. La RMSD entre nos solutions et les structures PDB est calculée par la méthode expliquée ci-dessus.

Lors de l'application de notre méthode, chaque molécule a été classifiée dans une famille pour obtenir un ensemble de paramètres. La table en annexe AT.4 présente les RMSD de chaque molécule de notre *ensemble de test* avec les paramètres de sa famille, et les paramètres que nous utilisons pour les autres familles. Nous pouvons voir que les ensembles de paramètres choisis s'adaptent bien à nos molécules par rapport aux autres ensembles. En effet, les RMSD pour la famille choisie sont plus faibles que pour les autres. Par exemple, pour 4QJH, la RMSD minimale atteinte est de 7.87 Å avec les paramètres préchoisis. La RMSD minimale passerait au-dessus de 10 Å si nous avions choisi les paramètres des autres familles. Notons aussi que, pour contrôle, notre approche a été évaluée sur l'*ensemble de référence* (voir la table en annexe AT.3).

La figure 4.11 montre le résultat de notre échantillonnage pour certaines molécules de l'*ensemble de test*. Pour des molécules simples, comme la molécule 1E8O, l'orientation globale des jonctions peut être retrouvée avec une précision raisonnable, même si nous n'avons pas d'information sur les interactions tertiaires.

Les paramètres choisis permettent d'avoir un échantillonnage qui donne une forme globale utilisable. Les RMSD calculées sont en moyenne en dessous de 10 Å. Pour la molécule 1MFQ (127 nucléotides), nous trouvons des structures avec une RMSD de 9.68 Å.

Nous observons grâce aux RMSD et aux visualisations que nos solutions représentent globalement la forme prise par les molécules. Nous observons aussi que les familles de 3-jonctions sont retrouvées (les hélices adjacentes aux 3-jonctions sont dans les bonnes directions).

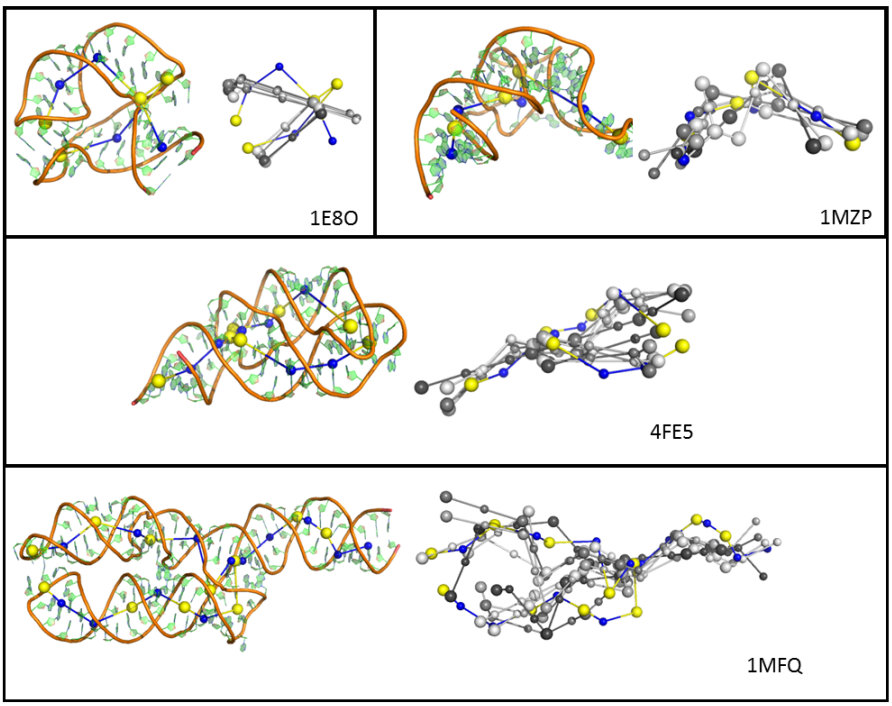


FIGURE 4.11 – **Visualisation des solutions proches de la structure PDB sur l'*ensemble de test*.** La structure trouvée est superposée à la structure PDB. À gauche, nous observons que notre représentation en graphe (en bleu et jaune) représente bien la structure *PDB*. À droite, nous superposons les cinq solutions les plus proches en gris (plus la couleur est sombre, plus la solution est proche) avec le graphe recherché. La gamme finale d'échantillon permet une vision globale du repliement, les 3-jonctions possédant les géométries recherchées, donc la bonne famille de repliement.

4.8 Comparaison avec les approches actuelles

Nous comparerons notre méthode avec plusieurs approches à grain fin existantes : iFoldRNA, MC-Sym, FARNA, et NAST. Nous ne comparerons pas les résultats de ces travaux préliminaires avec RNAComposer. Les méthodes à gros grain, RNAJAG et ERNWIN, n'utilisent pas les mêmes représentations que nous. Au vu de leur modèle, nous ne pouvons pas trouver de moyen de passer de leur représentation à la notre, il est donc difficile de comparer nos résultats aux leurs. Nous indiquerons la RMSD minimale trouvée par ces méthodes dans la littérature ou avec leur propre logiciel. Pour les autres approches, nous passerons leur résultat dans notre représentation pour calculer la RMSD par rapport à la structure PDB, comme indiqué dans la section 4.6.

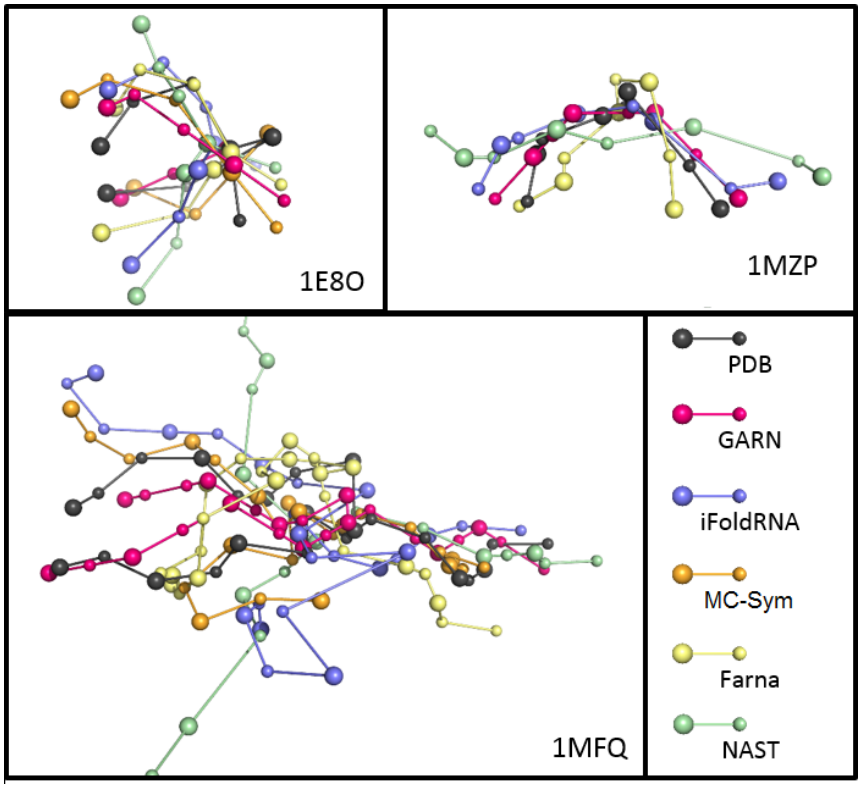


FIGURE 4.12 – **Visualisation pour une comparaison des différentes approches**. La meilleure solution de GARN (en rose) et ses équivalents à gros grain pour les solutions des autres méthodes sont superposées à la structure réelle (en noir). Visuellement, le repliement de GARN est celui qui représente le mieux la structure recherchée.

La table 4.2 fournit la RMSD de la structure la plus proche de la structure PDB et le nombre de structures proches de la solution pour l'*ensemble de test*. La table annexe AT.3 fournit les RMSD minimales pour les méthodes à grain fin sur l'*ensemble de référence*. La figure 4.12 montre la meilleure solution pour certaines de ces méthodes.

Un des avantages de GARN est qu'il ne nécessite pas de base de données de fragments ou d'informations sur les structures tertiaires. Certaines méthodes sont aussi limitées par la taille de la molécule, ce qui augmente leur coût de calcul. Les plus grandes molécules (plus de 100 nucléotides) peuvent être repliées par GARN, avec un temps de calcul rapide : environ une heure pour un échantillonnage de 50 molécules (sur un ordinateur Intel Core i5, 2.60GHz CPU, 6 GB RAM) pour la molécule 4TS0, contre 2h pour iFoldRNA (sur leur serveur), 8h pour MC-Sym (sur leur serveur), 4h pour FARNA (sur notre ordinateur), et 25 min pour NAST (sur notre ordinateur). Ce dernier n'opérant pas de repliement dû aux interactions tertiaires, son temps de calcul est rapide mais les RMSD de ses structures

ID	Nucl.	RMSD	GARN	iFoldRNA	MC-Sym	FARNA	NAST	RNAJAG	ERNWIN
1MZP	55	min	4.32	8.23	NA	5.61	12.97	6.74	4.09
		max	10.58	12.68	NA	17.80	26.38	–	–
# de RMSD < 5Å			*7*	*0*	–	*0*	*0*	–	–
1E8O	49	min	6.82	12.62	6.75	7.78	12.34	–	7.63
		max	15.40	17.71	15.55	21.02	20.29	–	–
# de RMSD < 8Å			*13*	*0*	*1*	*1*	*0*	–	–
4FE5	67	min	7.14	15.04	NA	7.80	17.00	–	4.75
		max	14.53	20.12	NA	20.98	34.17	–	–
# de RMSD < 9Å			*2*	*0*	–	*1*	*0*	–	–
4QJH	74	min	7.87	11.34	NA	8.99	23.44	–	–
		max	14.93	21.24	NA	18.67	26.57	–	–
# de RMSD < 9Å			*3*	*0*	–	*1*	*0*	–	–
4TS0	89	min	10.42	NA	NA	9.47	19.84	–	–
		max	23.13	NA	NA	23.19	22.79	–	–
# de RMSD < 12Å			*4*	–	–	*4*	*0*	–	–
1LNG	97	min	7.85	11.92	10.52	12.67	36.49	14.56	7.08
		max	17.07	35.53	29.58	30.19	59.66	–	–
# de RMSD < 10Å			*6*	*0*	–	*0*	*0*	–	–
4WFL	107	min	8.82	18.08	NA	11.33	43.41	–	13.63
		max	16.22	25.75	NA	26.00	47.04	–	–
# de RMSD < 10Å			*5*	*0*	–	*0*	*0*	–	–
4QK8	124	min	12.25	18.66	NA	13.23	54.43	–	13.87
		max	22.78	28.49	NA	22.19	59.44	–	–
# de RMSD < 14Å			*3*	*0*	–	*1*	*0*	–	–
1MFQ	127	min	9.68	20.42	16.07	16.13	38.91	10.55	11.91
		max	20.64	34.08	30.97	41.27	44.17	–	–
# de RMSD < 11Å			*7*	*0*	–	*0*	*0*	–	–

TABLE 4.2 – **Comparaison avec les autres méthodes.** Ce tableau compare GARN avec les structures fournies par iFoldRNA, MC-Sym, FARNA, NAST, RNAJAG et ERNWIN. Nous notons en bleu les RMSD minimales atteintes. Certains serveurs ne peuvent pas générer de solutions pour tout l'*ensemble de test*. iFoldRNA n'utilise que la séquence comme information d'entrée. Pour NAST, nous ne lui fournissons que la structure secondaire (car nous n'avons pas d'informations sur les interactions tertiaires). Ses informations sont nécessaires au bon repliement de NAST. Les résultats de RNAJAG proviennent de la littérature [Kim *et al.*, 2014]. Ceux de ERNWIN proviennent de leur logiciel qui fournit une structure d'énergie minimale. Notre méthode donne de bons résultats globaux pour les structures, mais fournit aussi plusieurs structures proches de ce minimal. Nous comptons le nombre de structures en dessous d'un certain seuil de RMSD pour observer le nombre de structures proches de la structure réelle sur tout l'échantillonnage de 50 structures.

sont élevées.

Si nous étudions le repliement des 3-jonctions, nous observons que la famille trouvée par le repliement global est la famille réelle de la molécule, en utilisant seulement l'information sur l'empilement. La figure en annexe AF.1 montre le résultat de la 3-jonction de la molécule 1MFQ. La table en annexe AT.5 montre les résultats des RMSD sur les joueurs des 3-jonctions de l'*ensemble de test*. GARN montre de bons résultats pour une méthode à gros grain, même face à des méthodes par fragments (comme FARNA). En effet, les RMSD sur les 3-jonctions sont moins bonnes que les autres méthodes tout en validant la famille de la jonction (donc la forme spatiale).

Nous avons aussi comparé d'autres 3-jonctions à la méthode RNAJAG [Kim *et al.*, 2014], qui s'est concentrée sur le repliement des jonctions. La table en annexe AT.6 détaille les résultats obtenus. Même si deux modèles n'utilisant pas le même graphe ne peuvent être directement comparés, nos résultats sont comparables à ceux de RNAJAG.

Alors que les méthodes classiques fonctionnent bien sur de petites molécules (comme sur l'*ensemble de référence*), GARN réussit à trouver de meilleurs échantillons à gros grain pour des molécules de plus de 50 nucléotides.

4.9 Conclusion

Nos résultats montrent que notre paradigme fonctionne sur de grandes molécules (environ 100 nucléotides), apportant une forme globale respectée, et des solutions souvent meilleures que les méthodes actuelles à grain plus fin.

Notre approche consiste à utiliser une modélisation à gros grains pour construire un jeu de repliement via des potentiels statistiques et des algorithmes de minimisation de regret. Cette première étude nous conforte dans l'idée que notre approche permet de trouver des structures d'ARN aussi correctes que les approches actuelles.

5 Analyse du jeu

Dans les chapitres précédents, nous avons vu qu'il était possible de représenter un repliement par un équilibre de Nash, et que notre premier modèle de jeu donnait des résultats prometteurs.

Nous allons étudier dans ce chapitre l'apport de la minimisation de regret de notre approche ainsi qu'une possibilité d'extension du modèle à des molécules possédant des 4-jonctions et plus.

La section 5.1 étudiera l'apport de la minimisation de regret en comparaison avec un algorithme de recherche d'équilibres ou une méthode de Monte-Carlo. La section 5.2 se focalisera sur la perspective d'une extension à des molécules possédant des 4-jonctions et plus, en observant les résultats sur une 4-jonction pour notre méthode actuelle. La section 5.3 étudiera les améliorations possibles de la méthode présentée dans le chapitre précédent.

5.1 Algorithmes de minimisation du regret

Nous utilisons des algorithmes de minimisation de regret pour atteindre un équilibre. Nous n'avons actuellement aucun moyen de démontrer qu'un équilibre de Nash a été atteint. Avec les algorithmes UCB et EXP3 (voir la section 2.3.3, page 21) que nous avons utilisés, nous avons cherché à minimiser le regret.

Lors de la phase d'apprentissage, chaque joueur peut ne pas avoir la possibilité de sélectionner toutes les stratégies (d'après les restrictions des stratégies). Nous sommes ici dans un contexte de *sleeping expert* [Freund *et al.*, 1997], où un *expert* peut choisir de passer un tour. Pour cela, nous adaptons la définition du regret cumulé, dans le sens où nous comparons notre résultat à la meilleure stratégie disponible au temps t. Nous

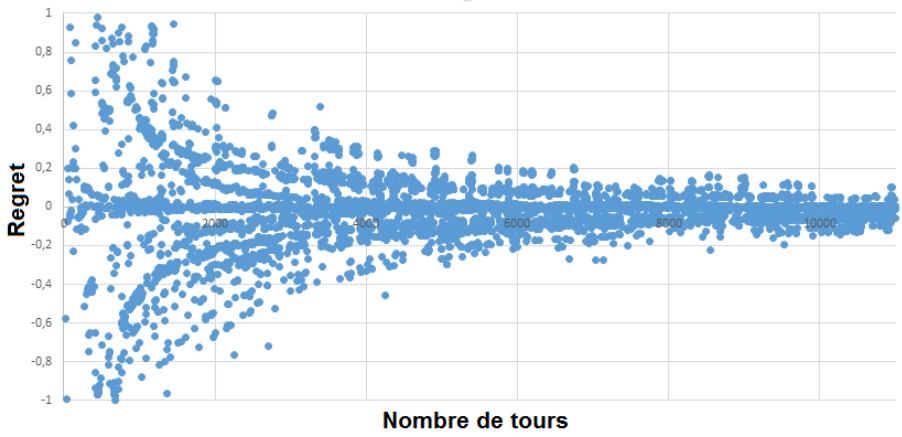


FIGURE 5.1 – **Évolution du regret cumulé moyen au fil du jeu pour un joueur.** Chaque point représente le regret pour un joueur d'une 3-jonction à chaque tour du jeu, durant le repliement de la molécule 4FE5. Pour certains tours de jeu et dans certaines configurations, la stratégie choisie permet au joueur d'avoir un meilleur gain que la stratégie optimale, donc un regret négatif. Après 4000 tours de jeu, l'amplitude du regret reste stable et presque nulle. Le joueur a ainsi minimisé son regret en considérant les stratégies des autres joueurs.

définissons ici le regret d'un joueur i comme :

$$Regret_i(T) = \max\left(\sum_{t=1}^{T} gain'_i(t)\right) - \sum_{t=1}^{T} gain_i(t) \tag{5.1}$$

où $gain_i(t)$ est le gain du joueur i au temps t d'après la stratégie choisie par notre algorithme et $gain'_i$ le gain du joueur i pour la meilleure stratégie possible et disponible au temps t.

Le regret peut prendre des valeurs positives et négatives, mais son amplitude devrait diminuer au fil du temps pour atteindre une valeur fixe correspondant à de légères divergences autour de la valeur nulle (le regret sera alors nul). Si nous regardons l'évolution du regret au fil du jeu (étape par étape) sur un joueur (voir la figure 5.1), le résultat montre que nous arrivons à faire tendre ce regret vers 0.

Comme vu dans la section 2.3.4, il est possible d'utiliser la minimisation de regret pour converger vers un équilibre de Nash corrélé. Notre jeu arrive à minimiser le regret, donc nous arrivons à nous rapprocher d'un équilibre de Nash corrélé. Sur nos résultats, nous atteignons une amplitude fixe, cohérente avec l'idée d'équilibre corrélé atteint.

5.1.1 Algorithme de théorie des jeux

Nous avons choisi d'approcher les équilibres de Nash grâce à des méthodes de minimisation de regret. Il existe des algorithmes cherchant directement des équilibres de Nash et tentant de converger vers ces derniers. L'algorithme LRI (*Linear Reward Inaction*) [Sastry *et al.*, 1994] (voir l'algorithme 2.1, section 2)est l'un d'eux. Nous tentons ici de voir l'effet de cet algorithme sur notre modèle de jeu.

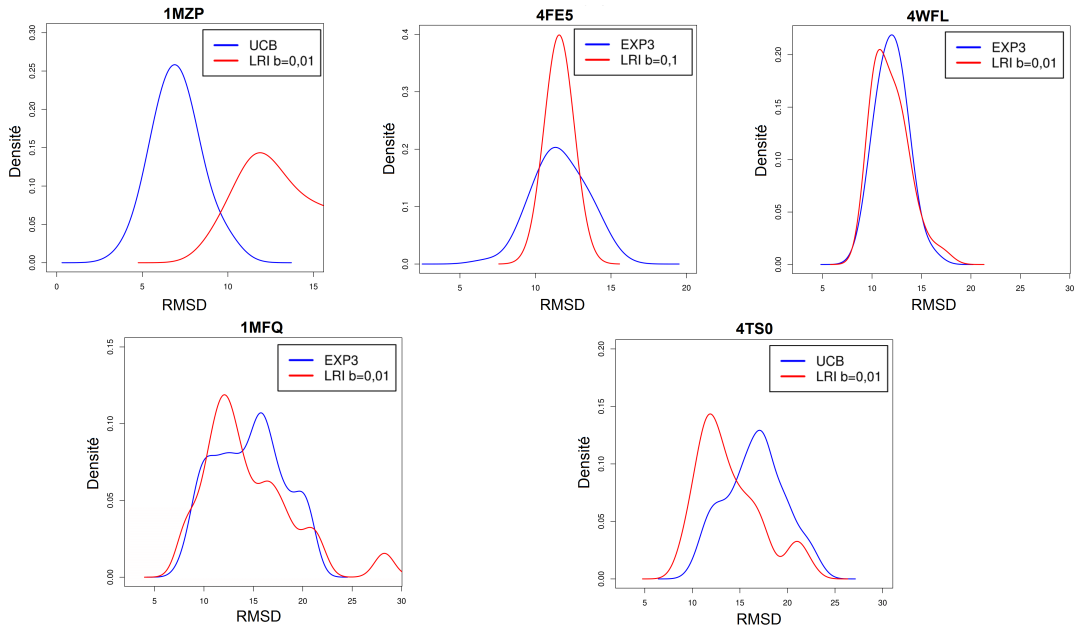


FIGURE 5.2 – **Distribution des RMSD avec des algorithmes de minimisation de regret et avec LRI.** Les structures pour un échantillonnage de 50 structures ont été calculées sur notre méthode avec des algorithmes de minimisation de regret (courbes bleues) et l'algorithme LRI (courbes rouges). Chaque graphe montre la densité (en ordonnée) des RMSD (en abscisse) pour une molécule donnée. La valeur du paramètre b est de 0.01. Dans le cas de 4FE5, b est de 0.1, permettant une convergence dans un temps convenable. Nous recherchons l'algorithme qui permet d'obtenir le plus de RMSD minimales.

Nous utilisons les mêmes paramètres de jeu qu'avec les algorithmes de minimisation de regret, mais nous remplaçons ces algorithmes par l'algorithme LRI (voir la figure 5.2 pour les résultats). L'algorithme LRI a été décrit dans la section 2.3.1. La valeur du paramètre b utilisée, permettant de calibrer la vitesse de convergence, est 0.01. Dans le cas de 4FE5, nous avons utilisé $b = 0.1$ pour accélérer la convergence. En effet, le temps de convergence de 4FE5 avec $b = 0.01$ ne nous permettait pas d'utiliser correctement l'algorithme (temps supérieur à 24h pour une solution). En ce qui concerne le temps de convergence, les deux algorithmes ont été testés sur exactement le même logiciel et le

même ordinateur, seul l'algorithme a changé. Nous avons observé que la recherche de 50 solutions pour la molécule 1MFQ prenait trois fois plus de temps avec LRI qu'avec EXP3.

L'algorithme LRI tente de converger vers un équilibre de Nash pur. Il peut converger mais nous ne pouvons pas vérifier que la solution finale soit un équilibre de Nash pur.

Dans la figure 5.2, les structures sont semblables (pour 4WFL ou 4FE5) ou meilleures (pour 1MZP ou 1MFQ) avec les algorithmes de minimisation de regret. Nous observons aussi que pour la molécule 4TS0 les structures trouvées par l'algorithme LRI sont plus proches de celle recherchée que les structures trouvées par l'algorithme UCB. Dans tous les cas, le temps de convergence de l'algorithme LRI est plus grand (trois fois plus long au minimum), pour des résultats globalement semblables, voir moins performants.

Nous ne retenons pas l'algorithme LRI. Nous travaillerons donc avec la minimisation de regret, pour approcher un grand nombre de bonnes solutions en un temps raisonnable (quelques minutes ou quelques heures).

5.1.2 Monte-Carlo

Les approches actuelles de prédiction de structures tridimensionnelles [Das et Baker, 2007, Parisien et Major, 2008], notamment les approches à gros grains [Kim *et al.*, 2014, Kerpedjiev *et al.*, 2015], utilisent la méthode de Monte-Carlo [Metropolis et Ulam, 1949, Metropolis, 1987]. Cette méthode utilise des tirages aléatoires pour calculer une quantité déterminée. La difficulté est de générer des conformations indépendantes avec une efficacité de l'algorithme acceptable. Si elles sont effectivement indépendantes, l'erreur sur les résultats diminue en proportion de l'inverse de la racine carré de $\mathcal{N}$. L'entier $\mathcal{N}$ permet de décider le nombre de fois que la stratégie s (la même stratégie que dans notre jeu) sera testée. Plus les stratégies seront testées, plus l'erreur diminuera.

Nous avons voulu comparer les solutions obtenues par un algorithme "classique" de Monte-Carlo et celles obtenues par nos algorithmes de minimisation de regret.

L'algorithme de Monte-Carlo (voir l'algorithme 5.1) va imposer une stratégie à un joueur, puis va tirer aléatoirement des stratégies pour les autres joueurs. Une fois la stratégie testée $\mathcal{N}$ fois, une nouvelle stratégie est testée. Nous choisissons ensuite la stratégie du joueur en prenant celle qui lui a permis de maximiser son gain. L'algorithme passe ensuite à un autre joueur.

Dans nos algorithmes actuels de minimisation de regret, nous effectuons *500 * N* tours (N étant le nombre de joueurs). Si nous considérons que les 12 stratégies possibles pour chaque joueur sont testées le même nombre de fois, chaque stratégie est donc testée environ $n_s = \frac{500*N}{12}$. Nous utiliserons ce nombre n_s comme variable $\mathcal{N}$ pour nos tests de

ALGORITHME 5.1 – **Méthode de Monte-Carlo**

Algorithme

```
1  Entrée : entier N
2  Sortie : ensemble de stratégies
3  pour chaque joueur j
4      pour chaque stratégie s de j
5          action[j] = s
6          pour i de 1 à N
7              pour chaque autre joueur k ≠ j
8                  action[k] = aleatoire()
               fin pour
           fin pour
       fin pour
9      action[j] = maximiseGain(action)
   fin pour
```

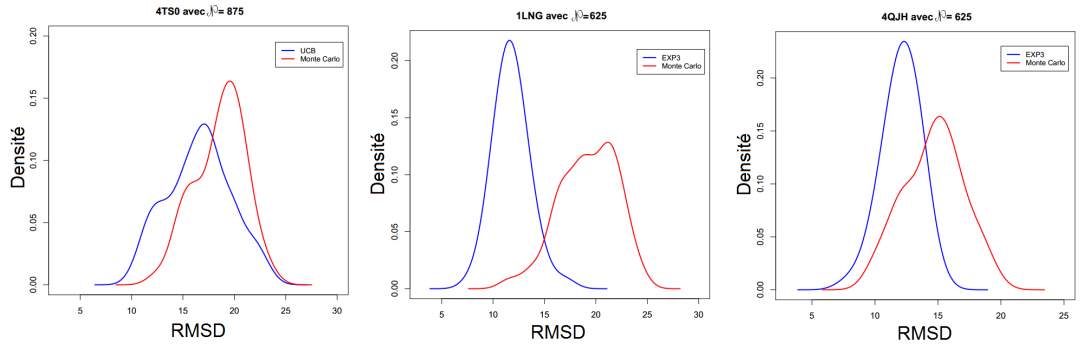


FIGURE 5.3 – **Distribution des RMSD avec des algorithmes de minimisation de regret et avec Monte-Carlo.** La courbe bleue indique la distribution des RMSD pour un échantillonnage de 50 structures avec l'algorithme UCB/EXP3. La courbe rouge indique la distribution des RMSD pour un échantillonnage de 50 structures avec l'algorithme de Monte-Carlo. Chaque graphe montre la densité (en ordonnée) des RMSD (en abscisse) pour une molécule donnée. Les échantillons ont été réalisés pour les molécules 4TS0, 1LNG et 4QJH. Les RMSD des structures calculées par UCB/EXP3 sont inférieures à celles trouvées par Monte-Carlo.

Monte-Carlo.

La figure 5.3 montre la densité des RMSD pour quelques molécules de notre *ensemble de test* du chapitre précédent, avec les algorithmes UCB/EXP3 et avec Monte-Carlo. Pour ces tests, aucun autre paramètre du jeu n'a été modifié. Nous observons que les RMSD des structures trouvées par minimisation de regret sont globalement plus faibles que celles trouvées par Monte-Carlo. La figure 5.4 montre la densité des RMSD pour une

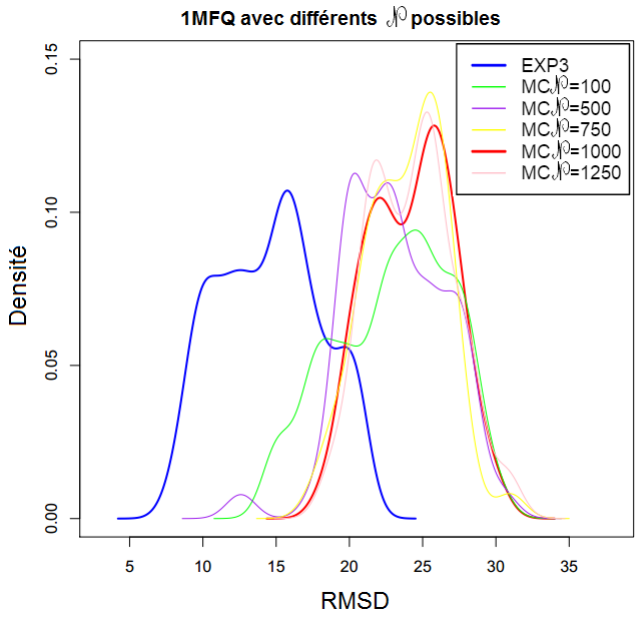


FIGURE 5.4 – **Distribution des RMSD avec des algorithmes de minimisation de regret et avec Monte-Carlo pour 1MFQ.** La courbe bleue indique la distribution des RMSD pour un échantillonnage de 50 structures avec l'algorithme EXP3. Les autres courbes indiquent la distribution des RMSD pour un échantillonnage de 50 structures avec l'algorithme de Monte-Carlo. Chaque couleur indique alors une valeur pour $\mathcal{N}$ différentes. La courbe rouge indique la valeur correspondant au nombre d'essais des stratégies concordant avec celui de GARN. Les RMSD des structures calculées par EXP3 sont inférieures à celles trouvées par Monte-Carlo.

molécule (1MFQ) en modifiant la valeur de $\mathcal{N}$. Quelle que soit cette valeur, l'algorithme EXP3 obtient des structures avec des RMSD plus faibles que presque toutes les structures obtenues avec Monte-Carlo.

Nous observons que l'échantillonnage par EXP3 ou UCB (les algorithmes utilisés dans GARN) obtient donc des RMSD plus faibles, donc plus proches des structures attendues, que la méthode de Monte-Carlo. Nous observons le même comportement si nous observons les résultats de notre approche présentée dans le chapitre suivant (voir la figure en annexe AF.3).

5.2 Passage au 4-jonctions

Nous avons appliqué notre méthode sur la molécule 4GXY. Cette molécule assez grande (172 nucléotides, 32 joueurs) possède une 4-jonction.

Lors de notre première étude de cette molécule, nous avions comme structure secondaire (récupérée via RNA FRABASE [Popenda *et al.*, 2010]) deux 3-jonctions plutôt qu'une 4-jonction. Nous avons alors modélisé cette molécule sans 4-jonction, mais avec deux 3-jonctions (voir la figure en annexe AF.2 pour la structure secondaire). La table 5.1 et la figure 5.5 montrent les résultats de notre approche sur une structure secondaire avec deux 3-jonctions pour représenter une 4-jonction.

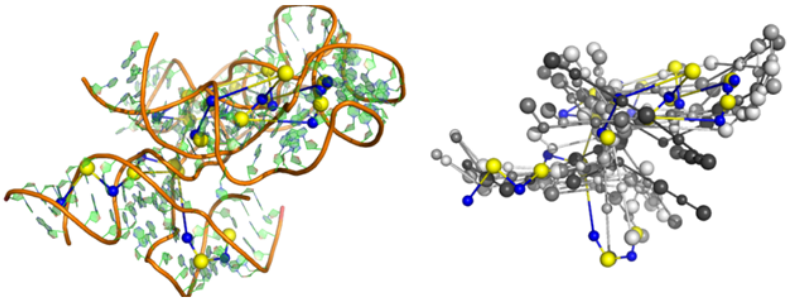


FIGURE 5.5 – **Visualisation des structures pour 4GXY.** La structure à gros grains PDB (en bleu et jaune) est superposée à sa structure PDB à grain fin. Nous superposons nos 5 meilleures structures en noir/gris à la structure à gros grain PDB. Nous observons que nos meilleures structures proposent des repliements de jonctions dans les bonnes directions par rapport à la structure PDB.

ID PDB	# Nucl.	RMSD	GARN	iFoldRNA	MC-Sym	FARNA	NAST	ERNWIN
4GXY	172	min	14.27	NA	NA	15.84	69.04	16.0
		max	31.04	NA	NA	26.17	74.83	–

TABLE 5.1 – **Résultats pour la molécule 4GXY avec des 3-jonctions.** Ce tableau compare les résultats de notre méthode (GARN) avec les approches actuelles. Les serveurs de MC-Sym et iFoldRNA ne renvoient pas de solution pour 4GXY. RNAJAG n'a pas de données pour 4GXY dans sa littérature. ERNWIN renvoie la solution d'énergie la plus basse. GARN permet de trouver des structures atteignant des RMSD plus basses que les autres approches actuelles. En bleu la RMSD minimale trouvée parmi toutes les méthodes.

Les résultats de cette molécule sont dans la même lignée que les autres molécules de l'*ensemble de test* du chapitre précédent : le résultat montre un bon repliement par rapport aux approches actuelles, et la forme globale de la molécule est proche de celle attendue. La 4-jonction arrive à se replier correctement en étant modélisé par deux 3-jonctions.

Ce résultat nous permet de considérer que notre modèle peut s'étendre à des molécules et des jonctions plus grandes.

5.3 Passage à la généralisation

Durant cette étude, nous avons mis en place un jeu permettant de replier des molécules d'ARN ne possédant qu'au plus des 3-jonctions. Ce travail préliminaire nous a permis d'avoir une idée sur les possibilités de notre modèle. Il reste de nombreux points à améliorer pour une généralisation du modèle plus complète et plus précise.

Nos futures améliorations concerneront par la suite :

L'utilisation ou non de la grille pour les distances entre les nœuds. Pour faciliter la modélisation, nous avons plongé notre graphe représentant la molécule dans une grille. La distance entre deux nœuds du graphe est ainsi limitée au nombre de pas d'une grille, contraignant fortement cette distance. Pour un même type d'élément de structure secondaire, par exemple une 2-jonction possédant 6 nucléotides et une autre 2-jonction en possédant 20 auront le même pas de grille. De plus, la distance calculée représente la distance entre un nœud et un autre, et non la place spatiale prise par le nœud. Dans la prochaine partie, nous travaillerons à améliorer la distance entre les nœuds en prenant en compte ces informations.

L'utilisation ou non de la grille pour les stratégies. Les stratégies des joueurs sont des directions d'après un repère spatial global sur la grille. Cela nous a permis de simplifier leurs calculs et leurs utilisations pour notre travail préliminaire. Ces stratégies peuvent être critiquées. En effet, en fonction des stratégies des autres joueurs, le choix d'une même stratégie (donc d'une direction dans l'espace) pour un joueur n'aura pas le même impact sur la structure. Par exemple, si le joueur 0 joue la stratégie numéroté 0 puis le joueur 1 joue la stratégie 0 aussi, alors les joueurs seront alignés. Mais si le joueur 0 joue la stratégie 1, alors la structure sera repliée, et le joueur 1 peut se retrouver dans la situation où la stratégie 0 n'est alors plus jouable. Lors de l'observation du repliement par notre méthode, nous voyons que les premiers joueurs (dans l'ordre de jeu) choisissent assez rapidement leur stratégie finale. Par conséquent, cela impose des directions aux joueurs suivants. Ces joueurs ne subissent alors que peu de changement de direction des joueurs précédents, car leurs stratégies n'ont que peu d'impact sur la valeur de leur fonction de gain. Il serait intéressant de se placer dans un repère local pour chaque joueur, pour que la stratégie choisie impacte mieux le joueur.
Enfin, la grille imposait des angles de 60° entre les stratégies. Il serait possible de s'extraire de ces contraintes pour proposer des angles plus fins.

L'ensemble de stratégies des 2-jonctions. Les molécules d'ARN possèdent un grand nombre de 2-jonctions. Dans le chapitre précédent, nous ne nous sommes pas focalisés sur la représentation des 2-jonctions. De plus, la limitation des angles due à l'utilisation de la grille ne rendait pas pertinent l'analyse des motifs des 2-jonctions. Pour d'autres paramètres (le gain et le pas de la grille), nous avions travaillé sur des statistiques pour les configurer. Il est alors possible de faire de même pour les stratégies des 2-jonctions.

Le calcul du gain pour les joueurs. Les potentiels permettant de calculer le gain sont configurés pour correspondre à des types d'élément de structure secondaire (hélice, 2-jonction, etc.). Nous allons étendre ces statistiques pour que ces potentiels puissent être utilisés quels que soient les éléments de structure secondaire.

La généralisation permettra de trouver des structures quelle que soit la taille de la molécule. Elle permettra aussi d'améliorer les premiers résultats que nous avons dans cette partie.

5.4 Conclusion

Nous avons utilisé des algorithmes de minimisation de regret et nous avons observé que le regret cumulé tend vers 0. Cette minimisation apporte un ensemble de structures plus proche du résultat attendu que la méthode de Monte-Carlo. Il est aussi plus intéressant d'utiliser des algorithmes comme EXP3 ou UCB que l'algorithme LRI car le temps de convergence de ce dernier peut être très long. Nous utiliserons donc dans la suite les algorithmes de minimisation de regret, que nous trouvons plus performants pour notre problème.

De plus, le modèle permet de s'étendre à de plus grandes molécules sans perdre de qualité par rapport aux autres approches actuelles.

Dans la partie suivante, nous mettrons en place les améliorations proposées dans ce chapitre en prenant en compte les connaissances acquises durant toute cette partie.

Partie III

Notre méthode de repliement de l'ARN

6 Généralisation du jeu

Nous avons obtenu dans les chapitres précédents un modèle permettant d'échantillonner les structures 3D possibles d'une molécule d'ARN. Cette méthode fonctionne sur des molécules possédant au plus une 3-jonction. Elle discrétise l'espace 3D et les possibilités de repliement de l'ARN. Nous étendons la méthode précédemment décrite afin de replier toute molécule d'ARN. Cette nouvelle approche prendra en compte les remarques que nous avons faites dans le chapitre précédent.

Dans la section 6.1, nous détaillerons la modélisation des joueurs, ainsi que leurs gains et stratégies. La section 6.2 expliquera la généralisation de notre jeu pour toutes les molécules d'ARN, notamment le choix des paramètres pour chaque molécule. Enfin, dans la section 6.3, nous analyserons nos résultats en quatre étapes. Tout d'abord, nous comparerons nos structures avec les structures recherchées. Puis nous observerons les structures que nous avons générées, en les comparant avec celles du chapitre précédent. Ensuite, nous les comparerons avec les méthodes actuelles de repliement d'ARN. Enfin, nous analyserons nos résultats pour des molécules possédant plus de 1500 nucléotides.

6.1 Modélisation de l'ARN

Dans le chapitre 4 du document, nous avons détaillé notre modélisation de la molécule d'ARN. Chaque élément de structure secondaire est vu comme un (ou plusieurs) nœuds d'un graphe. Les 1-jonctions et 2-jonctions, ainsi que les hélices courtes (moins de 5 paires de bases) sont représentées par un seul nœud, alors que les hélices longues et les 3-jonctions sont représentées par deux nœuds (ou plus pour les très longues hélices possédant plus de 10 paires de bases). Rappelons que nous avions considéré uniquement les molécules ayant au plus des 3-jonctions. Nous allons étendre notre modèle à toutes les tailles de jonctions.

6.1.1 Représentation des joueurs

La représentation de l'ARN se base sur celle du chapitre précédent (voir la section 4.1, page 39), c'est-à-dire que la molécule est représentée par un graphe. Ainsi, nous avons déjà des représentations correctes pour les hélices, 1-jonctions, 2-jonctions et 3-jonctions. Nous allons construire une représentation pour les 4-jonctions et plus.

Au sein du graphe, chaque joueur i possède un joueur précédent $i-1$ (qui décide de la position de i) et un joueur suivant (que le joueur courant i positionne dans l'espace). Pour harmoniser les représentations, nous modélisons les 4-jonctions et plus pour qu'elles ne soient constituées que de joueurs avec deux arêtes. Pour arriver à ce résultat, nous allons superposer plusieurs joueurs sur la même position (sur un même nœud). La figure 6.1 décrit la représentation d'une jonction (quelle que soit sa taille), et la figure 6.2 présente la modélisation pour une molécule avec une 6-jonction.

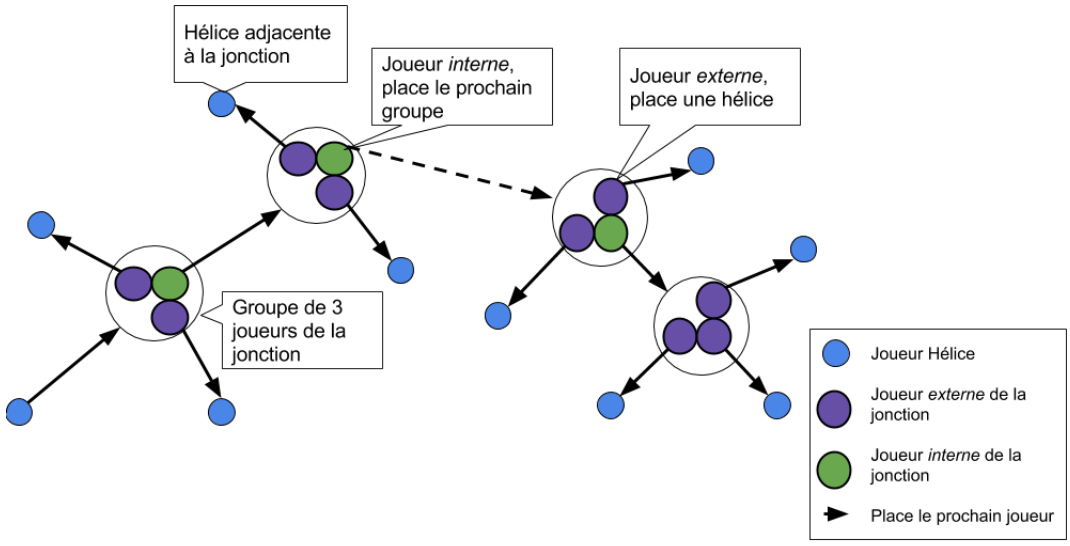


FIGURE 6.1 – **Représentation des jonctions.** Nous modélisons ici un exemple de représentation générique d'une jonction. Chaque joueur possède un joueur précédent et un joueur suivant. Certains joueurs *externes* (en violet) donneront la direction (indiquée par une flèche) des hélices adjacentes (en bleu), et d'autres joueurs *internes* (en vert) la direction de la jonction en plaçant dans l'espace le prochain nœud (groupe de joueurs) de la jonction. Un nœud (grand cercle contenant le groupe de joueurs) peut contenir 2 ou 3 joueurs.

Les grandes jonctions (4-jonction et plus) sont représentées par deux types de joueurs : les joueurs *externes*, qui placeront dans l'espace les hélices liées à la jonction, et les joueurs *internes*, qui placeront les autres joueurs de la jonction. Pour positionner facilement les joueurs dans notre espace, nous superposons jusqu'à 3 joueurs sur un même nœud

du graphe. Chaque nœud possédera au minimum deux joueurs *externes*, et le troisième joueur pourra, s'il existe, être un joueur *interne* ou *externe*. Il n'y aura que des joueurs *externe* sur le dernier nœud de la jonction (dans l'ordre du graphe) : deux joueurs si la jonction est de taille impaire, trois joueurs sinon. Les autres nœuds de la jonction posséderont deux joueurs *externes* et un joueur *interne*. Chaque joueur contrôle toujours la direction d'une unique arête dans le graphe. L'ordre ce ces joueurs est toujours l'ordre 5'-3' (sens de synthèse des acides nucléiques).

Soit une jonction de taille T_j. Les joueurs *externes* (J_e) positionnent les hélices adjacentes qui ne sont pas encore positionnées, donc $T_j - 1$ hélices, car la première hélice (dans l'ordre du graphe) est déjà placé par rapport à la jonction. Il y a donc $T_j - 1$ joueurs *externes*. Les joueurs *internes* (J_i) positionnent les autres nœuds de la jonction. Chaque nœud contenant au minimum deux joueurs *externes* (le dernier nœud contient deux ou trois joueurs *externes* d'après la parité de T_j). Il y a donc $\lfloor (|J_e|/2) \rfloor$ nœuds à placer. Sachant que le dernier nœud ne contient pas de joueur *interne*, il y aura donc $\lfloor (|J_e|/2) \rfloor - 1$ joueurs *internes*. Le nombre de joueurs pour une jonction j de taille T_j est alors : $T_j + \lfloor (T_j - 1)/2 \rfloor - 2$.

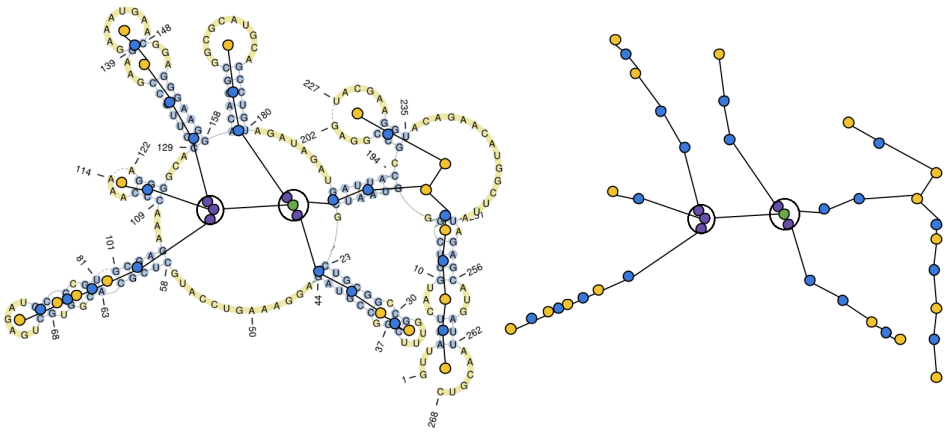


FIGURE 6.2 – **Graphe de la molécule 3DHS.**Pour une meilleure visibilité du graphe, nous présentons ici le graphe de la molécule 3DHS avec et sans sa structure secondaire. Chaque joueur possède un joueur précédent et un joueur suivant. Certains joueurs *externes* (en violet) donneront la direction des hélices adjacentes, et d'autres joueurs *internes* (en vert) la direction de la jonction. Les joueurs dans le même cercle seront sur le même nœud du graphe. Les hélices sont représentées en bleu et les jonctions en jaune. La 6-jonction est représentée par deux nœuds du graphe et 6 joueurs dans le jeu.

Chaque nœud de la jonction (regroupement de deux ou trois joueurs) correspond au barycentre des paires de bases auxquelles ils sont associés et des nucléotides de la jonction. Ce calcul de la position des nœuds par rapport aux nucléotides est aussi appliqué aux

1-jonctions et 2-jonctions. Cette modélisation garde une représentation simple du graphe (deux arêtes pour chaque joueur), et prend en compte la place prise spatialement par la jonction. Cela est généralisable à des jonctions de taille arbitraire sans introduire un trop grand nombre de joueurs. De plus, nous gardons ainsi le même système de jeu : un joueur place une arête (et donc un autre nœud) dans l'espace.

6.1.2 Distance entre les nœuds

La grille utilisée précédemment imposait des distances entre les nœuds multiples de 5.6 Å (voir la section 4.1.2, page 42). Nous allons nous rapprocher des distances réelles des molécules.

Rappelons que les arêtes ont une longueur représentant la distance spatiale entre deux nœuds. Nous introduisons la notion de *largeur* d'un nœud, qui va correspondre à la largeur spatiale de l'élément de structure secondaire du nœud. Cette largeur peut être vue comme le diamètre d'une sphère contenant l'élément de structure secondaire. La distance entre deux nœuds adjacents sera la somme de la moitié de leur largeur.

Nous allons calculer la distance de deux façons différentes : en utilisant la grille précédemment décrite (largeur *statistique*), ou en considérant la distance entre deux paires de bases (largeur *par nucléotide*).

Cette largeur sera un paramètre du jeu et aura donc un impact sur le jeu.

Largeur *statistique*.

Les largeurs *statistiques* sont les distances calculées dans la section 4.1.2 du chapitre 4 (page 42). La largeur des nœuds est alors de 5.6 Å, ou de 11.2 Å pour les hélices longues (représentées par deux nœuds ou plus) ou pour les grandes 2-jonctions (possédant plus de 6 nucléotides). Cette largeur nous permet de reprendre la notion de discrétisation de l'espace vue dans la section 4.1.2.

Largeur *par nucléotide*.

La distance entre deux paires de bases de l'hélice d'ARN, (connue pour être le plus souvent une hélice de forme « A ») est de 2.4 Å [Ussery, 2002]. Nous calculons ainsi la largeur des nœuds pour chaque élément (voir Figure 6.3) :
- **Hélices** : la largeur du nœud sera le nombre de paires de bases multiplié par 2.4 Å ;
- 1**-jonctions** : la largeur du nœud sera la moitié du nombre de nucléotides de la jonction multipliée par 2.4 Å ;
- 2**-jonctions** : nous considérerons que le plus petit brin non apparié est dans la continuité des hélices adjacentes, alors que l'autre brin se replie dans l'espace. Nous choisissons de calculer la largeur de la jonction par rapport au plus petit brin. La

largeur du nœud représentant la 2-jonction sera le nombre de nucléotides sur le plus petit brin non apparié multiplié par 2.4 Å. Si le plus petit brin ne possède pas de nucléotide, nous considérerons qu'il en possède un seul (pour ne pas avoir de largeur nulle) ;

– 3**-jonctions** : pour le nœud représentant la famille de jonction (voir la figure 4.2 qui décrit les joueurs des 3-jonctions), nous comptons le nombre de nucléotides sur le plus grand brin non apparié multiplié par 2.4 Å. Pour le nœud représentant l'empilement, la largeur sera le nombre de nucléotides sur le plus petit brin non-apparié multiplié par 2.4 Å ;

– 4**-jonctions et plus** : la largeur du nœud est le nombre total de nucléotides de la jonction divisé par la taille de la jonction, le tout multiplié par 2.4 Å.

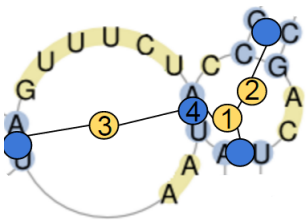


FIGURE 6.3 – **Exemple de calculs pour les largeurs *par nucléotide*.** Nous représentons ici une partie de la structure secondaire d'une molécule. Les joueurs 1 et 2 font partis d'une 3-jonction. Le joueur 1 aura pour largeur le nombre de nucléotides sur le plus petit brin non-apparié multiplié par 2.4Å. Ici, le plus petit brin possède 0 nucléotide. Dans ce cas-là, nous considérons qu'il y en a un, sa largeur sera donc de $1 * 2.4$ Å. Le joueur 2 aura pour largeur le nombre de nucléotides sur le plus grand brin non apparié multiplié par 2.4 Å, donc $2 * 2.4$Å ici, soit 4.8Å. Le joueur 3 représente une 2-jonction, et sa largeur dépendra de son plus petit brin non apparié, donc $2 * 2.4 = 4.8$Å ici. Le joueur 4 représente une hélice avec une seule paire de base, sa largeur sera donc de $1 * 2.4$Å.

6.1.3 Stratégies des joueurs

Les stratégies d'un joueur permettent de choisir la position du nœud (du joueur ou groupe de joueur) suivant dans le graphe. En choisissant la position du nœud suivant, le joueur crée un angle particulier entre ses deux arêtes adjacentes et donc un repliement dans l'espace. Chaque nœud représentant un élément de structure secondaire ou une partie de l'élément, un joueur ne pourra pas former tous les repliement possibles, il sera donc important de limiter les angles possibles, et donc les stratégies. Les joueurs choisiront leur stratégie les uns après les autres, comme dans le chapitre 4. L'ordre des joueurs dépend de la jonction de taille la plus élevée (d'après l'ordre 5'-3'), puis suit l'ordre 5'-3' (voir la figure 4.3, page 41). Pour les joueurs présents dans les grandes jonctions (4-jonctions

ou plus), pour un même nœud, les joueurs *externes* jouent (d'après l'ordre 5'-3') puis le joueur interne.

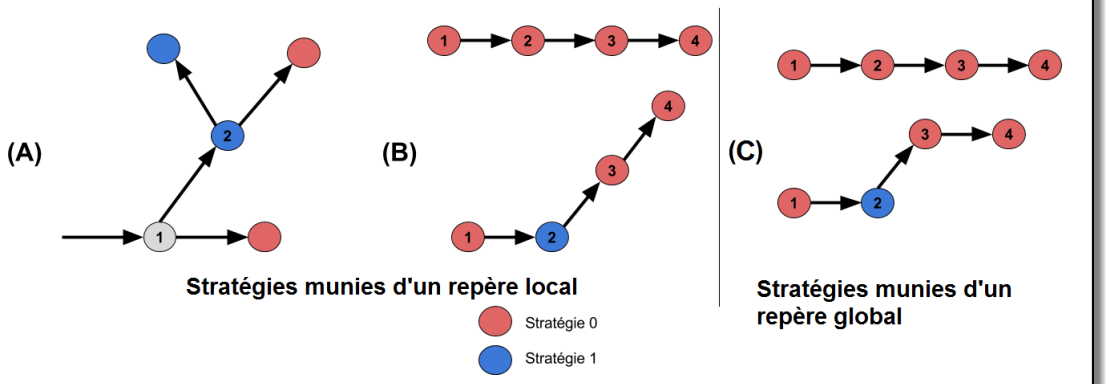


FIGURE 6.4 – **Choix et impact des stratégies *munies d'un repère local*.** Les cercles représentent les joueurs, et les flèches représentent les directions vers lesquelles un joueur peut en positionner un autre. Chaque stratégie est représentée par une couleur différente. La figure (A) montre deux stratégies possibles (0 et 1) pour le joueur 1 et le joueur 2. La figure (B) montre le changement de stratégie pour le joueur 2. Dans le premier cas, le joueur 2 a choisi la stratégie 0 (rouge) et suit ainsi la stratégie du joueur précédent. Dans le second cas, le joueur 2 a choisi la stratégie 1 (bleue) et modifie ainsi la structure de la molécule. La figure (C) rappelle l'impact des stratégies pour les stratégies *munies d'un repère global*, comme dans la figure 4.9 du chapitre 4.

Les stratégies peuvent être vues comme :

- des stratégies *munies d'un repère global* dans l'espace. Ces stratégies correspondent aux stratégies décrites dans le chapitre 4. Dans ce cas-là, les stratégies correspondent à des directions par rapport à un repère spatial global.
- des stratégies *munies d'un repère local* dans l'espace. Ces stratégies prendront en compte plus précisément les angles entre les joueurs. Dans ce cas-là, les stratégies correspondent à des directions par rapport à l'arête précédent le joueur : cette arête indiquera une direction au joueur lui permettant de créer un repère local pour ses stratégies.

Stratégies *munies d'un repère global*. Comme dans le chapitre précédent, la direction prise par un joueur dépend d'un ensemble de stratégies indiquant les directions dans l'espace d'après le repère spatial ((+x,0,0), (-x,0,0), (0,0,+z), etc.). Dans une grille triangulaire 3D, il existe 12 directions possibles, donc 12 stratégies pour chaque joueur. Ces stratégies sont celles utilisées dans le chapitre 4. La figure 4.9 du chapitre 4 (page 46) décrit l'effet de ces stratégies sur la structure d'une molécule. Sur ces 12 stratégies,

le joueur aura à chaque tour des restrictions sur ses stratégies d'après la stratégie du joueur précédent et d'après son type d'élément de structure secondaire.

Stratégies *munies d'un repère local*. Les stratégies *munies d'un repère local* se baseront sur les stratégies *munies d'un repère global* mais prendront directement en compte la stratégie du joueur précédent et les restrictions qui lui sont imposées. Les stratégies dépendront d'un repère local dont la première direction sera la même direction que celle du joueur précédent (les autres se placeront toujours de la même façon autour de cette direction). Les autres directions spatiales seront alors calculées d'après ce nouveau repère. Les stratégies possibles pour un joueur seront alors les directions (dépendantes du joueur précédent) qui lui sont autorisées. La figure 6.4 décrit l'effet de ces stratégies sur la structure d'une molécule.

Avec l'utilisation d'un *repère local*, les restrictions sur les stratégies se font en amont, ce qui donne un ensemble de stratégies inférieur à 12 stratégies pour un joueur. En effet, la stratégie impliquant un angle à 180° est de base retirée pour éviter les superpositions. De plus, si le joueur ne peut pas former un angle supérieur à 60° (par exemple une hélice), son ensemble de stratégies contiendra 5 stratégies (en nous basant sur la grille triangulaire 3D) au lieu de 12. Nous pouvons alors profiter de cette réduction de l'ensemble de stratégies pour ajouter d'autres stratégies : les stratégies avec un angle de 30°.

L'ajout de stratégies permettant des angles de 30° confèrent une plus grande flexibilité des joueurs. Si nous ajoutions des angles à 30° dans le cas du *repère global*, l'ensemble de stratégies deviendrait trop grand (42 stratégies). Avec un nombre trop élevé de stratégies, le temps de convergence pour un joueur augmenterait fortement.

Ces nouvelles stratégies nous entraînent à reconsidérer les stratégies des joueurs, notamment des 2-jonctions.

Stratégies des 2-jonctions.

Nous allons améliorer la qualité du repliement des 2-jonctions. En effet, ces jonctions étant présentes dans la plupart des molécules, leur repliement est important pour la structure globale. Nous proposons dans ce chapitre un ensemble de stratégies basé sur des statistiques de repliement des molécules de notre *ensemble de référence* (disponible dans la table annexe AT.1).

Nous étudions l'angle formé par les 2-jonctions (l'angle entre les deux arêtes du joueur) en fonction du nombre de nucléotides de la jonction. Les résultats sur notre *ensemble de référence* sont présentés dans la figure 6.5.

D'après ces résultats, nous traiterons l'ensemble de stratégies des 2-jonctions de deux façons différentes : si le plus petit brin non-apparié contient 0 nucléotide, l'ensemble de stratégies sera entre 30° et 120°. Dans les autres cas, il sera entre 0° et 60°. Cet ensemble

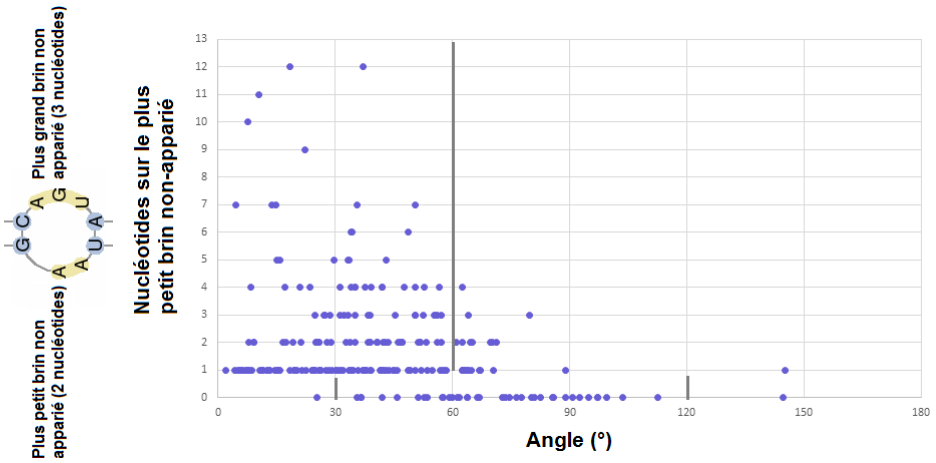


FIGURE 6.5 – **Angles des 2-jonctions d'après le nombre de nucléotides sur le plus petit brin non apparié.** Le plus petit brin non apparié correspond au brin d'une 2-jonction possédant le moins de nucléotides. Nous observons une corrélation entre le nombre de nucléotides sur le plus petit brin non-apparié et l'angle formé par la jonction. Plus le nombre de nucléotides sur ce brin est élevé, plus le joueur est restreint à des angles de moins de 60°. Les traits gris correspondent aux limitations des angles que nous utiliserons dans notre méthode.

sera l'ensemble *statistique* pour les 2-jonctions. Dans le cas des stratégies *munies d'un repère local*, les restrictions se feront comme indiqué ci-dessus. Dans le cas des stratégies *munies d'un repère global*, les restrictions se feront seulement sur des angles entre 60° et 120°, ces stratégies ne possédant pas d'angle à 30°.

Il existe des motifs 3D pour les 2-jonctions (voir la figure 1.8 de l'état de l'art, page 9), mais la limitation de nos angles (0°, 30° ou 60° entre deux positions) ne nous permet pas de les utiliser correctement. Comme pour les familles des 3-jonctions (voir la figure 1.9 de l'état de l'art, page 9), nous laissons les joueurs choisir la famille correcte, après avoir restreint légèrement leurs possibilités d'après les nucléotides sur le plus petit brin non apparié.

Stratégies des hélices.

Dans le cas des stratégies *munies d'un repère global*, les restrictions sont les mêmes : les hélices restent droites si elles sont petites (inférieures à 6 paires de base), sinon elles peuvent se replier de 60°. Dans le cas des stratégies *munies d'un repère local*, nous ajoutons plus de souplesse : les hélices de moins de 6 paires de base ont le droit à 30° de déviation, 60° sinon.

Nous observons toujours le cas où les hélices peuvent être *rigides* (restreintes à 0° de

déviation).

Stratégies des 3-jonctions et plus.

En ce qui concerne les 3-jonctions et les jonctions plus larges, les joueurs n'ont pas de restrictions spécifiques. Dans le cas des stratégies en *munies d'un repère global*, les stratégies sont les mêmes qu'au chapitre 4.

Il existe des familles de repliement 3D pour les 4-jonctions (voir la figure 1.10 de l'état de l'art, page 10), mais la limitation de nos angles (0°, 30° ou 60° entre deux positions) ne nous permet pas de les utiliser correctement. Notons aussi que plusieurs familles de 4-jonctions sont identiques dans notre représentation à gros gain (comme la famille H et π, ou la famille χ et ψ).

6.1.4 Gain des joueurs

Le gain d'un joueur sera, comme dans le chapitre 4 (voir la section 4.3, page 48), la somme de potentiels entre lui et tous les autres joueurs. Les potentiels du chapitre 4 étaient calculés d'après des statistiques sur notre *ensemble de référence* (sans la molécule "1Z58"). Nous avons moins d'informations sur les 4-jonctions (moins de dix 4-jonctions dans notre ensemble pour 76 molécules), et utiliser la même méthode ne permettrait pas d'avoir des potentiels pertinents pour ces jonctions. Il en est de même pour les jonctions plus larges (5-jonctions et plus).

Pour cela, nous simplifions ces potentiels en calculant leurs paramètres d'après le type global du joueur : hélice ou jonction (voir la figure 6.6). Nous avons alors pour chaque potentiel trois ensembles de paramètres possibles : entre deux hélices, entre deux jonctions, ou entre une hélice et une jonction. Cela permet de généraliser nos potentiels.

Nous testerons alors plusieurs gains :
- en utilisant le potentiel de Lennard-Jones, de Lennard-Jones seuillé, ou de Gauss (voir la section 4.3, page 48) ;
- les paramètres de ces potentiels dépendront des types des joueurs (hélice, 2-jonction, etc.) noté *typé* ou des types globaux des joueurs (hélice ou jonction) noté *H/J*. Dans le cas des 4-jonctions et plus, nous considérerons toujours les types globaux des joueurs.

6.2 Fonctionnement du jeu

Nous avons toutes les informations utiles pour notre jeu : qui sont les joueurs, quelles sont leurs stratégies et comment calculer leur gain. Nous allons étudier le fonctionnement global de notre jeu et les paramètres que nous allons utiliser.

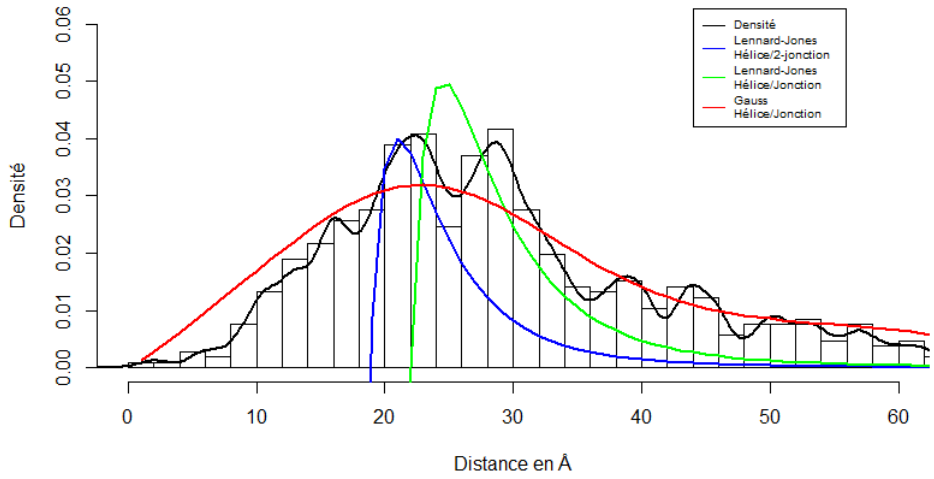


FIGURE 6.6 – **Potentiels entre une hélice et une jonction.** En se basant sur les statistiques des distances entre les hélices et les jonctions (en noir), nous calculons un potentiel de Lennard-Jones (en vert) et un potentiel de Gauss (en rouge). Le potentiel de Lennard-Jones impose une unique distance optimale, ici entre les deux pics (à 23 Å et à 28 Å) pour n'en pénaliser aucun. En effet, en plaçant le potentiel de Lennard-Jones sur l'un des pics, l'autre aurait eu un potentiel nul voir négatif. Avec le potentiel représenté ici, les deux pics ont des potentiels environ égaux, et non nuls ou négatifs. En bleu le potentiel précédemment calculé, entre une hélice et une 2-jonction. La courbe bleue sera donc utilisée pour un potentiel *typé*, et la courbe verte pour un potentiel *H/J*.

Nous allons choisir dans un premier temps quel algorithme et quelle séquence nous allons utiliser dans ce chapitre (voir la section 4.4, page 51). Nous étudierons ensuite les différents paramètres présentés ci-dessus.

6.2.1 Calcul d'équilibre et algorithme

Dans la partie précédente, nous avons étudié plusieurs méthodes de calcul d'équilibre possibles ainsi que deux algorithmes (EXP3 et UCB). Deux possibilités sont ressorties : utiliser une méthode de calcul (OA) (un joueur joue puis tous les joueurs mettent à jour leurs probabilités) avec l'algorithme UCB ou une méthode de calcul (AA) (tous les joueurs jouent puis tous les joueurs mettent à jour leurs probabilités) avec l'algorithme EXP3. Pour les molécules possédant une 3-jonction, les paramètres (AA/EXP3) étaient utilisés, sinon nous utilisions (OA/UCB) pour les molécules ne possédant que des 2-jonctions et moins. Les paramètres (OA/UCB) fonctionnent bien pour des molécules qui avaient un espace de solutions de structures plus petit. Avec les nouveaux paramètres introduits, il y aura plus de repliements possibles (notamment grâce à la refonte des ensembles de

stratégies) ce qui nous permettra d'avoir de meilleurs résultats avec (AA/EXP3). Pour globaliser notre jeu, nous utiliserons ici seulement les paramètres (AA/EXP3).

6.2.2 Stratégies des joueurs

Pour étudier l'impact de nos paramètres, nous allons travailler sur notre *ensemble de référence* (voir la table en annexe AT.1). Pour cela, chaque paramètre sera testé sur cet ensemble en utilisant toutes les possibilités pour les autres paramètres non testés. Nous observerons ainsi les différences sur le paramètre qui nous intéresse, sachant que tous les autres paramètres ont aussi été globalement testés dans ce cas là.

Nous cherchons des paramètres utilisables sur le plus grand nombre de molécules, mais aussi assez spécifiques pour pouvoir replier au mieux les structures. Nous allons observer, pour chaque paramètre, la densité des RMSD normalisées (les RMSD sont normalisées de la plus petite à la plus grande pour les échantillons obtenus) de toutes les molécules de l'*ensemble de référence*. Cela nous permettra d'avoir une vision globale de son impact et de choisir une valeur unique pour le paramètre. Si nous observons, en analysant chaque molécule, que le choix unique n'est pas assez bon pour toutes les molécules, nous les étudierons une à une pour en sortir une règle permettant de différencier chaque cas. Cette méthode d'étude sera appliquée pour tous les paramètres testés.

Nous testons d'abord deux ensembles de stratégies différents : les stratégies *munies d'un repère global* et les stratégies *munies d'un repère local*.

Stratégies *munies d'un repère global* ou *munies d'un repère local*.

Sur notre *ensemble de référence*, nous observons que l'utilisation des stratégies *munies d'un repère local* permet d'obtenir plusieurs structures avec des RMSD plus faibles (voir la figure 6.7) et donc des structures plus proches des structures réelles.

Nous avons testé les stratégies *munies d'un repère local* pour améliorer les structures trouvées, en permettant de diminuer le nombre de stratégies à tester pour certains joueurs, et en donnant plus d'efficacité aux stratégies choisies. Notre observation (la figure 6.7) est donc en accord avec notre hypothèse d'une amélioration avec les stratégies *munies d'un repère local*.

Nous utiliserons dans notre approche les stratégies *munies d'un repère local* pour toutes les molécules.

Stratégies des deux jonctions. Nous testons l'apport des stratégies *statistiques* des 2-jonctions sur notre *ensemble de référence*. Nous supposons qu'elles améliorent le repliement global de nos molécules.

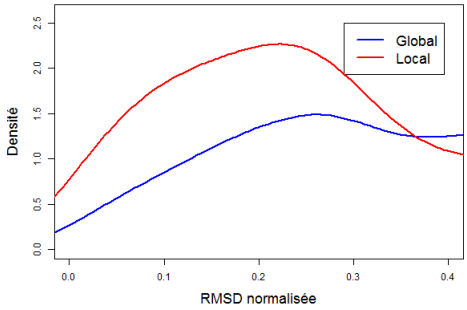


FIGURE 6.7 – **Densités des RMSD d'après le paramètre des stratégies *munies d'un repère global* ou *munies d'un repère local*.** Les densités des RMSD faibles (inférieures à 40% de la RMSD maximale) des structures utilisant les stratégies *munies d'un repère global* (en bleu) ou *munies d'un repère local* (en rouge) permettent d'observer une préférence pour les stratégies *munies d'un repère local*. En effet, les RMSD faibles des structures des molécules de notre *ensemble de référence* trouvées par ces stratégies sont plus nombreuses que celles trouvées par les stratégies *munies d'un repère global*.

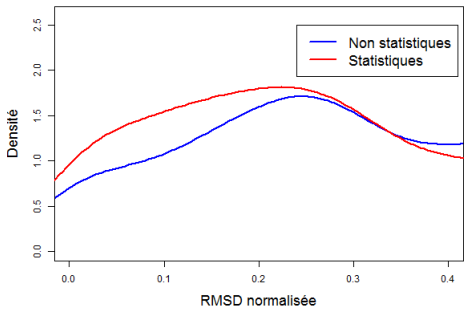


FIGURE 6.8 – **Densités des RMSD d'après le paramètre des stratégies des 2-jonctions.** Les densités des RMSD faibles (inférieures à 40% de la RMSD maximale) des structures utilisant les stratégies de la partie précédente (en bleu) ou les stratégies *statistiques* de ce chapitre (en rouge) pour les 2-jonctions montrent des RMSD faibles plus nombreuses pour la courbe rouge.

En utilisant ces stratégies dans notre jeu, nous observons (voir la figure 6.8) une amélioration des RMSD trouvées. En effet, la densité des RMSD faibles est plus importante avec les stratégies *statistiques*.

L'utilisation de statistiques nous permet ici d'améliorer nos résultats. L'ensemble de stratégies *statistiques* sera utilisé pour les 2-jonctions.

Hélices *rigides*.

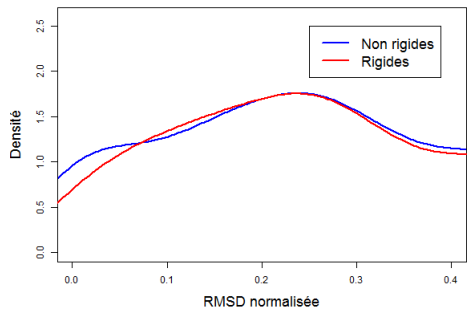


FIGURE 6.9 – **Densités des RMSD d'après le paramètre hélice *rigide* ou non.** Les densités des RMSD faibles (inférieures à 40% de la RMSD maximale) des structures utilisant les hélices *rigides* (en rouge) ou non (en bleu) ne sont pas très différentes. Nous ne pouvons statuer sur l'utilisation ou non des hélices *rigides*.

Dans le chapitre précédent, nous avions testé la rigidité des hélices. Certaines molécules (celles sans 3-jonction) se repliaient mieux avec des hélices rigides. Nous avons modifié dans certains cas les stratégies des joueurs, ce qui peut changer fortement l'impact des hélices *rigides*. Nous avons testé l'effet des hélices *rigides* sur notre *ensemble de référence*. En effectuant la même étude que sur les autres paramètres, nous observons sur la figure 6.9 que nous ne pouvons nous décider clairement entre la rigidité ou non des hélices. Le choix pour ce paramètre se fera donc en fonction de l'analyse des molécules elles-mêmes.

Nous observons que certaines molécules se replient mieux avec des hélices rigides. Par exemple, en étudiant la molécule 2G91, nous observons que, pour notre échantillonnage, la RMSD minimale atteinte dans le cas des hélices non rigides est de 4.38 Å. Dans le cas des hélices rigides, la RMSD minimale monte à 8.75 Å. Inversement, pour la molécule 1X8W, la RMSD minimale atteinte est de 22.25 Å dans le cas des hélices rigides et de 28.07 Å dans le cas des hélices non rigides. Nous observons donc des différences nettes pour certaines molécules. Nous avons recherché la cause possible de ces préférences.

En analysant les molécules, nous observons que la forte présence de jonctions influe sur la rigidité des hélices. Soit N le nombre de joueurs et N_{3+} le nombre de joueurs de type 3-jonction ou plus. En observant notre ensemble d'après le ratio $\frac{N}{N_{3+}}$, nous observons que les hélices rigides fonctionnent mieux lorsqu'il y a un petit nombre de jonctions (de N_{3+}).

Nous supposons, d'après notre étude des molécules de l'*ensemble de référence*, que s'il y a peu de jonctions ($\frac{N}{N_{3+}}$ élevé), alors le repliement de ces quelques jonctions est très important pour la forme globale. La direction prise par ces jonctions devra alors se prolonger par les hélices. Dans ce cas-là, les hélices devront être rigides.

Dans le cas où il n'y a pas de 3-jonction ou plus, nous observons que les hélices non rigides permettent un meilleur repliement. Sachant qu'il y a presque autant de 2-jonctions que d'hélices dans ces molécules, il y a donc une forte présence de jonctions, ce qui correspond à notre observation sur la rigidité des hélices et l'influence des jonctions.

Après observation, nous plaçons le ratio entre hélices rigides et non rigides à $\frac{N}{N_{3+}} = 5$.

En résumé, les hélices seront :
- non *rigides* si les jonctions sont assez nombreuses dans la molécule par rapport au nombre de joueur. Ce sera le cas pour les molécules ne possédant pas de 3-jonctions ou plus, et pour celle dont $\frac{N}{N_{3+}} < 5$;
- *rigides* si les jonctions ne sont pas assez nombreuses dans la molécule par rapport au nombre. Ce sera le cas pour les molécules dont $\frac{N}{N_{3+}} \geq 5$.

6.2.3 Largeurs des joueurs

Nous avons développé deux largeurs possibles : une largeur *statistique* et une largeur *par nucléotide*. La largeur *par nucléotide* s'éloigne de la discrétisation forte de la grille, en n'imposant plus des distances multiples de 5.6 Å entre les joueurs. Nous allons tester si le passage à la largeur *par nucléotide* est bénéfique à notre repliement.

La figure 6.10 montre les différences de densité de RMSD pour l'ensemble des molécules de l'*ensemble de référence*. Il est difficile de choisir une largeur particulière pour l'ensemble des molécules.

Comme précédemment, nous allons analyser les molécules une à une pour trouver une approche permettant de choisir entre les deux largeurs. Nous observons ici de grandes différences de RMSD dans nos structures pour plusieurs molécules. Par exemple, pour la molécule 259D, la RMSD minimale de notre échantillon est de 1.43 Å avec la largeur *statistique* alors qu'elle est de 5.15 Å avec la largeur *par nucléotide*. Inversement, la molécule 1NUJ montre une préférence pour la largeur *par nucléotide*, avec une RMSD minimale de 1.21 Å contre 4.51 Å avec la largeur *statistique*. Nous observons donc des différences nettes pour certaines molécules. Nous avons recherché la cause possible de ces préférences.

La largeur *par nucléotide* dépend de la longueur entre deux paires de bases, qui forment les hélices. Nous avons donc observé la présence d'hélices dans nos molécules. Soit n_{nu} le nombre de nucléotides et n_h le nombre d'hélices. En observant notre ensemble de

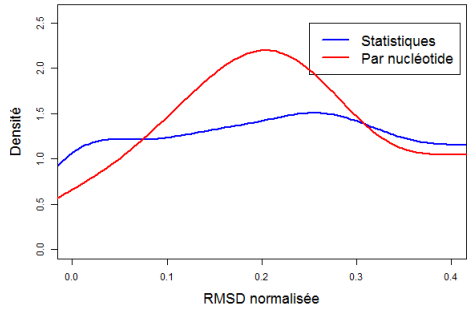


FIGURE 6.10 – **Densités des RMSD d'après le paramètre de largeurs.** Les densités des RMSD faibles (inférieures à 40% de la RMSD maximale) des structures utilisant les largeurs *statistiques* (en bleu) ou les largeurs *par nucléotide* (en rouge) proposent deux échantillons différents. La courbe bleue montre un premier pic sur des RMSD plus faibles que pour la courbe rouge. La courbe rouge a un premier pic sur des RMSD moins faibles que la courbe rouge, mais plus nombreuses. Il est difficile de décider entre les deux largeurs.

molécules d'après le ratio $\frac{n_{nu}}{n_h}$, nous observons que la forte présence des hélices (par rapport aux nucléotides) permet une utilisation efficace de la largeur *par nucléotide*, alors que l'inverse montre une préférence pour la largeur *statistique*.

Dans le cas où la présence des hélices est importante ($\frac{n_{nu}}{n_h}$ faible), de nombreux nucléotides de la molécule sont présents dans les hélices. Nous présageons que l'importance des hélices est alors assez élevée pour que la largeur *par nucléotide* convienne bien. Dans le cas inverse ($\frac{n_{nu}}{n_h}$ élevé), il n'y a pas assez de nucléotides présents dans les hélices. Nous utiliserons la largeur *statistique*, qui permet après observation de mieux représenter les structures dans notre modèle. Les 2-jonctions, qui peuvent fortement changer de largeur d'une jonction à l'autre, sont mieux représentées par la largeur *statistique*, qui est moins précise mais plus globale. Dans le cas des molécules sans 3-jonction ou plus, les 2-jonctions sont des joueurs très présents et très importants. Nous utiliserons uniquement la largeur *statistique*, qui semble ne pas provoquer des erreurs trop importantes sur la largeur des 2-jonctions.

En résumé, la largeur utilisée sera :
- la largeur *statistique* s'il n'y a pas une présence assez importante de nucléotides au sein des hélices. Ce sera le cas pour les molécules sans 3-jonction ou plus, ou dont le rapport $\frac{n_{nu}}{n_h} \geq 13$;
- la largeur *par nucléotide* s'il n'y a une présence assez importante d'hélices. Ce sera le cas pour les molécules dont le rapport $\frac{n_{nu}}{n_h} < 13$.

Nous avons placé la différence entre largeur *statistique* et *par nucléotide* à $\frac{n_{nu}}{n_h} = 13$ après des observations sur les molécules de l'*ensemble de référence*.

6.2.4 Gain des joueurs

Nous avons développé plusieurs potentiels pour le gain des joueurs. Nous avions notamment utilisé, dans la partie précédente, des potentiels Lennard-Jones ou Lennard-Jones seuillé d'après les familles que nous avions formées pour les molécules. La figure 6.11 montre les différences de densité de RMSD pour l'ensemble des molécules de l'*ensemble de référence* d'après ces potentiels.

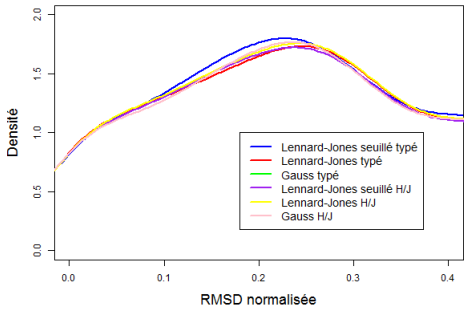


FIGURE 6.11 – **Densités des RMSD d'après les différents potentiels.** Les densités des RMSD faibles (inférieures à 40% de la RMSD maximale) des structures d'après les différents potentiels montrent qu'il n'est pas possible de préférer un potentiel pour toutes les molécules.

La densité globale ne nous donnant aucune information, nous allons étudier les molécules pour pouvoir choisir le potentiel.

Le potentiel permet de choisir le comportement de repliement des joueurs, et donc la forme globale de la structure. Nous observons, en analysant les résultats molécule par molécule, des préférences de potentiels pour certaines molécules.

Dans tous les cas, nous regardons la RMSD minimale atteinte sur nos échantillons de structures d'après les potentiels étudiés. Dans le cas de la molécule 2A2E par exemple, la RMSD minimale atteinte est de 16.31 Å avec le potentiel de Gauss *typé*, contre 17.42 Å avec le potentiel de Lennard-Jones *H/J* (et des RMSD encore plus élevées pour les autres potentiels). Par contre, pour la molécule 1KXK, la RMSD minimale atteinte est de 3.73 Å avec le potentiel de Lennard-Jones *H/J* contre 4.98 Å avec le potentiel de Gauss *typé*, ou encore 5.19 Å avec le potentiel de Lennard-Jones *typé* et 4.4 Å avec le potentiel de Lennard-Jones seuillé *H/J*. La molécule 1D4R préfère se replier avec le potentiel de Lennard-Jones *typé*, qui lui permet d'atteindre une RMSD minimale de 3.32 Å, contre minimum 4.2 Å pour les autres potentiels. Autre exemple, la molécule 1KH6, qui a pour RMSD minimal 5.11 Å avec le potentiel de Lennard-Jones seuillé *H/J*, alors qu'elle n'atteint par exemple que 7.13 Å avec le potentiel de Gauss *typé*.

D'après la structure secondaire de la molécule, nous pouvons présumer d'un repliement important ou non de la structure 3D. Dans une même branche de notre graphe d'ARN (parties composée d'hélice et de 2-jonction), la présence d'hélices longues (possédant plus de 5 paires de bases) provoquera des structures plus allongées, ces dernières se repliant peu. La présence des 2-jonctions dans ces mêmes branches provoquera des structures plus repliées.

Nous savons aussi que le type de potentiel impactera le repliement. Pour comparer l'impact des différents potentiels sur le repliement, nous analysons d'abord l'apport du type de potentiel (Gauss, Lennard-Jones ou Lennard-Jones seuillé). Le potentiel de Gauss permet une plus grande liberté pour les joueurs, l'écart-type de la fonction de Gauss étant plus élevé que celui de Lennard-Jones. Notons aussi que le potentiel de Lennard-Jones donnera moins de liberté que le potentiel de Lennard-Jones seuillé. En effet, le seuil implique une pénalité moins forte si le repliement n'est pas correct : si la distance entre deux joueurs est trop faible, le gain entre les deux joueurs est de 0 pour le Lennard-Jones seuillé et est négatif pour le Lennard-Jones.

Nous pouvons ensuite étudier l'impact des potentiels *typés* ou *H/J*. En effet, entre les potentiels *typés* ou *H/J*, nous observons un décalage du pic principal de quelques Å de plus pour le potentiel *H/J*. Par conséquent, les potentiels *typés* induisent un repliement plus compact que les potentiels *H/J* (la distance optimale étant à une distance plus faible).

De ces conclusions et de l'étude des molécules, nous pouvons choisir les potentiels. Nous observons plusieurs cas :

- Lorsque la molécule possède un grand nombre de 2-jonctions, le potentiel de Gauss permet un espace de solutions (de structures 3D) assez large pour que les 2-jonctions puissent tester tous les repliements possibles sans une pénalité trop forte. Ces jonctions peuvent alors tester plus facilement plusieurs repliements. Pour garder une molécule compacte, le potentiel de Gauss *typé* donnera de bons résultats ;
- La présence importante d'hélices longues (représentées par deux joueurs ou plus) va impliquer une structure plus longiligne que repliée. Par conséquent, plus le nombre d'hélices longues est important, plus le potentiel doit forcer le repliement pour contre balancer le faible repliement des hélices. Soit N le nombre de joueurs et N_{hl} le nombre d'hélices longues.
  - Si les hélices longues sont nombreuses, alors la meilleure façon d'imposer un repliement fort est d'utiliser le potentiel de Lennard-Jones *typé* ;
  - S'il y a peu ou pas d'hélices longues, alors le potentiel de Lennard-Jones seuillé *H/J* permet d'imposer un repliement moins compact ;
  - Pour un entre-deux, le potentiel de Lennard-Jones *H/J* permet de replier les molécules en imposant une compacité moins contraignante.

En résumé, les potentiels utilisés seront :

– Si la molécule possède sept 2-jonctions ou plus, alors le potentiel sera Gauss *typé* ;
– Si le rapport $\frac{N}{N_{hl}}$ est inférieur ou égal à 7, alors le potentiel sera Lennard-Jones *typé* ;
– Si le rapport $\frac{N}{N_{hl}}$ est entre 7 et 20, alors le potentiel sera Lennard-Jones *H/J* ;
– Si le rapport $\frac{N}{N_{hl}}$ est supérieur ou égal à 20, alors le potentiel sera Lennard-Jones seuillé *H/J*.

Nous avons placé la différence entre les potentiels d'après nos observations des molécules de l'*ensemble de référence* et de notre analyse des potentiels.

Cette analyse de notre *ensemble de référence* nous permet d'avoir une méthode automatique de repliement que nous allons tester sur un nouvel ensemble de test.

Dans le chapitre précédent, nous avions différencié les ensembles de paramètres en famille de molécules. Dans ce chapitre, nous avons différencié chaque paramètre d'après notre connaissance des structures secondaires. Regrouper les paramètres utilisés en familles n'aurait pas permis à certaines molécules de se replier au mieux.

6.2.5 Apport de la discrétisation

Dans la partie précédente, nous avons discrétisé l'espace de solutions des structures. Dans ce chapitre, nous avons testé plusieurs paramètres visant à s'éloigner de cette discrétisation.

En particulier, les paramètres sur les largeurs des joueurs (*statistiques* ou *par nucléotide*) permettaient soit de s'approcher d'un espace très discrétisé (*statistiques*) soit plus continu (*par nucléotide*). Nous avons développé le paramètre des largeurs *par nucléotide* pour améliorer les distances entre les joueurs. La largeur *statistique* permet d'avoir des largeurs moyennes pour chaque joueur, d'après des statistiques. Les largeurs *par nucléotide* calculent des distances plus précises entre les joueurs. Notre modélisation de l'ARN étant à gros grain, la largeur *statistique* est un bon compromis entre précision (grâce aux statistiques) et généralisation.

Sur une modélisation à gros grain, nous considérons qu'il est parfois plus intéressant de ne pas redescendre à un grain plus fin (au niveau du nucléotide) pour les paramètres de notre modèle. Notre représentation à gros grain est une représentation qui généralise tout un élément de structure secondaire en un point. Si nous souhaitons être plus précis en ajoutant des informations valides à grain fin, nous modifions alors notre modèle avec des informations imprécises à gros gain.

6.3 Résultats

Pour évaluer notre méthode, nous utiliserons la méthode d'évaluation présentée dans la section 4.6, page 54. L'ensemble de test utilisé dans la partie précédente ne contenait pas

des molécules avec des 4-jonctions ou plus. Nous allons donc étudier notre méthode sur un *ensemble de test élargi* (voir la table en annexe AT.7). Cet *ensemble de test élargi* contient toutes les molécules de *ensemble de test* de la partie précédente ainsi que des molécules plus grosses (jusqu'à 268 nucléotides). Chaque molécule de cet *ensemble de test élargi* sera comparé avec les structures présentes dans la base de données *Protein data bank*, les structures PDB.

6.3.1 Résultats de notre méthode

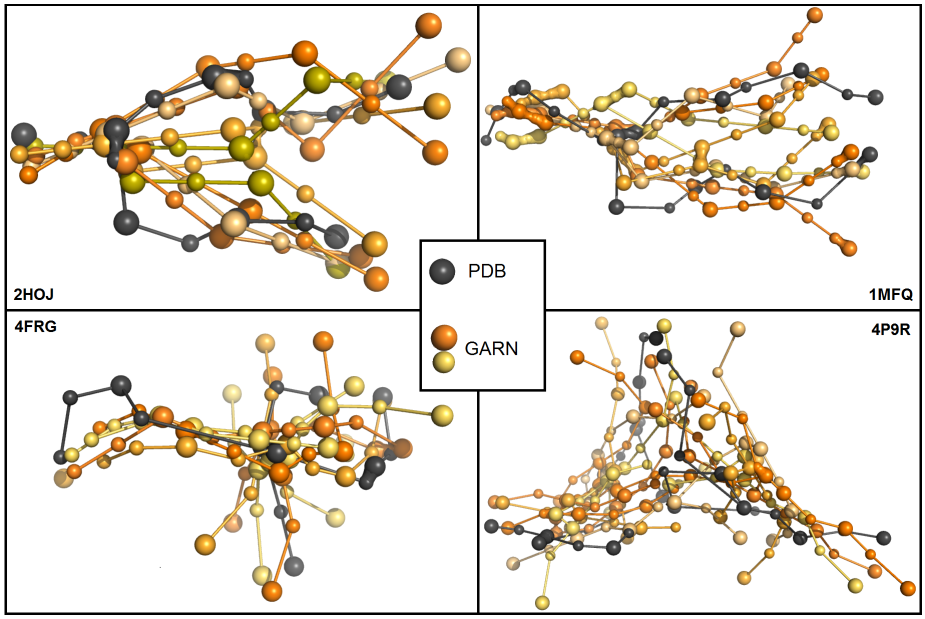


FIGURE 6.12 – **Visualisation des structures proches de la structure PDB pour des molécules de l'*ensemble de test élargi*.** Nous superposons les structures les plus proches en orange (plus la couleur est sombre, plus la structure est proche) avec le graphe de la structure PDB recherché (en noir). La gamme finale d'échantillon permet une vision globale du repliement, la jonction principale possédant la géométrie recherchée.

Nous avons testé sur notre *ensemble de test élargi* présenté ci-dessus notre méthode, GARN, avec les paramètres que nous avons définis dans la section précédente. La figure 6.12 présente les cinq meilleures structures par rapport à la structure PDB que nous souhaitons atteindre. La table 6.1 indique les valeurs des RMSD trouvées pour chaque molécule de notre *ensemble de test élargi*.

Nous observons grâce aux RMSD et aux visualisations que nos structures représentent globalement la forme prise par les molécules, notamment au niveau des jonctions.

ID PDB	# Nucl.	# Joueurs	RMSD			
			Minimum	1^{er} quartile	Moyenne	Maximum
1E8O	50	8	5.80	7.41	8.87	13.02
1MZP	55	8	4.20	5.95	6.98	11.03
4FE5	68	14	6.43	10.03	12.27	18.85
2DU3	71	13	10.14	13.72	14.87	20.54
4QJH	74	15	6.44	9.12	12.01	18.90
3Q3Z	77	9	10.24	11.51	13.82	18.91
1P5O	77	19	6.90	8.70	11.26	20.72
2HOJ	79	15	6.62	9.43	12.31	19.11
4FRG	84	17	7.99	12.80	14.37	19.83
4TS0	89	21	7.47	10.39	11.17	15.47
2GIS	94	21	10.72	12.62	14.07	17.13
1LNG	97	16	8.45	15.30	18.27	28.08
4WFL	107	18	8.93	11.69	13.64	19.52
1C2X	120	17	11.45	15.33	17.79	25.12
4QK8	124	20	13.79	17.14	19.41	24.50
1MFQ	127	24	9.62	17.33	20.35	30.32
1GID	158	27	16.36	26.04	30.08	37.72
3D0U	161	32	12.40	20.09	22.38	29.08
2QBZ	161	28	13.00	22.14	25.76	36.69
4GXY	172	34	14.75	21.08	23.23	29.41
4P8Z	188	29	14.60	20.01	21.65	27.55
4P9R	192	30	14.23	20.49	22.30	30.55
4GMA	210	36	21.33	27.22	28.66	34.17
4C4Q	233	49	16.32	24.68	27.07	39.43
3DHS	268	42	16.54	21.51	23.11	28.75

TABLE 6.1 – **Résultats sur l'ensemble de test élargi.** Pour chaque molécule de notre *ensemble de test élargi*, nous avons calculé 50 structures d'après notre méthode GARN expliquée dans ce chapitre. Ce tableau présente les résultats des RMSD des structures en comparaison avec la structure *PDB*. Nous avons calculé la RMSD minimum, moyenne et maximale, ainsi que son 1^{er} quartile.

Pour des molécules inférieures à 100 nucléotides, nous arrivons à atteindre des RMSD minimales inférieures ou égales à 10 Å. Pour des molécules supérieures à 100 nucléotides, nous arrivons à garder des RMSD minimales entre 10 et 20 Å.

6.3.2 Comparaison avec la méthode préliminaire

Durant ce chapitre, nous avons étendu notre méthode de repliement, choisissant des paramètres différents par rapport à notre méthode préliminaire du chapitre 4. Notre

méthode préliminaire a été développé pour des molécules ne possédant qu'au plus des 3-jonctions. La figure 6.13 montre la différence d'échantillonnage entre notre méthode préliminaire et notre méthode actuelle pour GARN. Dans cette figure, nous comparons les échantillonnages de 50 structures pour chaque molécule de l'*ensemble de test* (que nous avons utilisé dans les chapitres 4 et 5). Nous observons que l'échantillonnage de notre méthode actuelle permet d'avoir des RMSD plus faibles et donc des repliements de meilleure qualité.

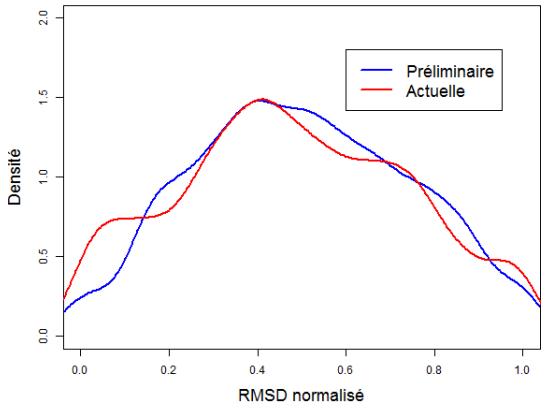


FIGURE 6.13 – **Densité des RMSD entre la méthode préliminaire et la méthode actuelle.** La courbe en bleu représente la densité des RMSD pour notre *ensemble de test* avec la méthode préliminaire du chapitre 4. La courbe en rouge montre la densité des RMSD avec la méthode actuelle. Nous observons que le premier pic de la courbe rouge se positionne sur des RMSD plus faibles que la courbe bleue. Par conséquent, les RMSD minimales trouvées par la méthode actuelle sont plus faibles que celles de la méthode préliminaire.

La méthode présentée ici est donc une meilleure approche globale de repliement des molécules que la méthode préliminaire. Pour des molécules comme 4TS0, la différence entre les RMSD minimales est de 3 Å entre la méthode préliminaire et celle actuelle (voir la figure 6.14), en faveur de la méthode actuelle. Cette différence de RMSD peut être plus faible, comme dans le cas de 1MZP avec de 0.12 Å de différence. Dans de rares cas, la différence entre les RMSD minimales est faible mais est en faveur de la méthode préliminaire. Par exemple, pour 1LNG, la méthode préliminaire atteignait 7.85 Å de RMSD minimale alors que celle actuelle atteint 8.45 Å (voir la figure 6.14).

Nous garderons comme méthode finale pour GARN celle présentée dans ce chapitre.

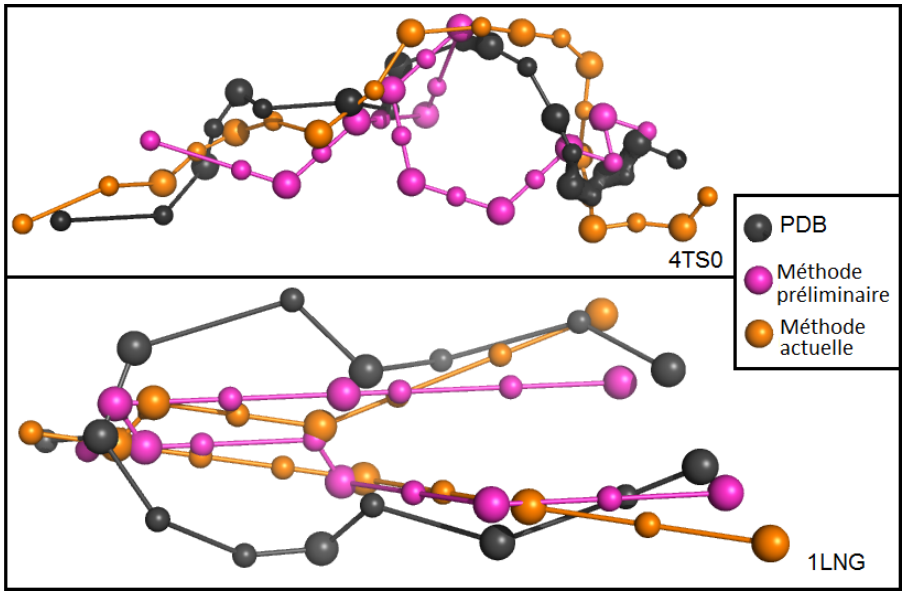


FIGURE 6.14 – **Visualisation des meilleures structures pour la méthode préliminaire et la méthode actuelle.** Nous visualisons la meilleure structure trouvée par la méthode préliminaire (en rose) et celle de la méthode actuelle (en orange). Pour la molécule 4TS0, la structure orange permet de retrouver correctement la structure noire (la structure PDB). Pour 1LNG, les deux structures peuvent être correctes, mais la structure rose dévie moins des directions prises par la structure noire.

6.3.3 Comparaison avec les logiciels existants

Il existe plusieurs approches existantes avec lesquelles nous allons nous comparer : FARNA, iFoldRNA, NAST, SimRNA, RNAComposer, ERNWIN et RNAJAG. Dans la partie précédente, nous nous sommes comparés avec MC-Sym. Sur notre *ensemble de test élargi*, nous n'avons pu tester que trois molécules avec MC-Sym, ce dernier ne calculant parfois que quelques solutions (moins de 10). Nous retirons donc ici la comparaison avec MC-Sym. Nous comparerons les résultats d'autres approches (FARNA, iFoldRNA, NAST, SimRNA, RNAComposer et ERNWIN), avec ceux de GARN, en utilisant les structures renvoyées par ces approches, comme présenté dans la section 4.6(page 54).

Les méthodes à gros grain (ERNWIN et RNAJAG) n'utilisent pas les mêmes représentations que nous, il est donc difficile de comparer nos résultats aux leurs. Nous utiliserons donc les RMSD présentées en l'état dans leurs publications (pour RNAJAG) ou les RMSD calculées par leur logiciel (pour ERNWIN).

Nous présentons ici les résultats de RNAComposer. Notons que ce dernier reconstruit un modèle 3D en utilisant la structure secondaire et le dictionnaire de RNA FRABASE

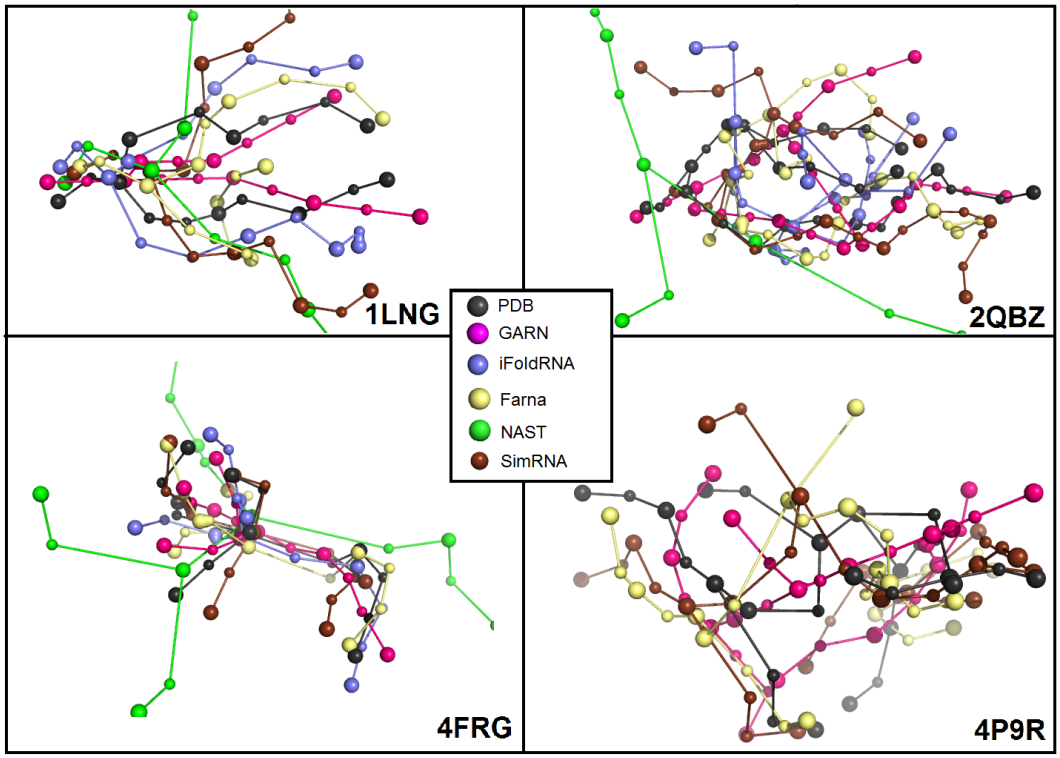


FIGURE 6.15 – **Visualisation des structures proches de la structure PDB pour des molécules de l'*ensemble de test élargi* calculées par GARN et par les autres approches existantes.** Nous superposons la structure la plus proche de la structure PDB (en noir) pour chaque approche. GARN (en rose) permet d'avoir une structure proche de la structure PDB. NAST (en vert) n'obtient que des structures très dispersées.

[Popenda *et al.*, 2010]. Il utilise ensuite la base de données pour récupérer des fragments 3D d'après la structure secondaire. Sachant que nous utilisons la même base de données de structures secondaires que RNAComposer, ce dernier peut directement récupérer des fragments des molécules testées, voire la molécule entière. Ses résultats seront donc bien meilleurs que les autres approches grâce à l'utilisation des données des fichiers de la base de données PDB, et donc un accès à la structure finale de la molécule (ou une partie de la structure finale s'ils ont besoin de plusieurs fragments pour la molécule).

Sur notre *ensemble de test élargi*, nous observons de meilleures RMSD avec GARN qu'avec les autres approches. La figure 6.16 montre l'échantillonnage de chaque méthode pour toutes les molécules de notre *ensemble de test élargi*. La table en annexe AT.8 regroupe les résultats des RMSD des structures pour toutes les approches sur notre *ensemble de test élargi*.

ID PDB	# Nucl.	GARN	FARNA	NAST	ERNWIN	iFoldRNA	RNACo.	SimRNA
1E8O	50	1 min.	∼ 5 min.	∼ 10 min.	∼ 20 min.	∼ 1 h 30	2 min.	∼ 5 h
4FE5	68	6 min.	∼ 4 h	∼ 20 min.	∼ 30 min.	∼ 2 h 30	3 min.	∼ 9 h
4QJH	74	6 min.	∼ 4 h30	∼ 20 min.	–	∼ 2 h	2 min.	∼ 9 h
2HOJ	79	7 min.	∼ 4 h	∼ 25 min.	∼ 40 min.	∼ 2 h	3 min.	∼ 8 h
4FRG	84	9 min.	∼ 4 h	∼ 25 min.	∼ 45 min.	∼ 5 h 40	3 min.	∼ 3 h
4WFL	107	9 min.	∼ 4 h 30	∼ 25 min.	∼ 1 h 20	∼ 3 h	4 min.	∼ 8 h
1MFQ	127	∼ 30 min.	∼ 5 h	∼ 25 min.	∼ 1 h	∼ 4 h 30	5 min.	∼ 15 h
4P8Z	188	∼ 40 min.	∼ 33 h	∼ 40 min.	∼ 4 h 40	–	7 min.	∼ 18 h
4GMA	210	∼ 1 h 30	∼ 45 h	∼ 45 min.	∼ 3 h 30	–	8 min.	–
3DHS	268	∼ 2 h 10	∼ 70 h	∼ 50 min.	∼ 4 h 15	–	12 min.	–

TABLE 6.2 – **Temps de calcul.** Ce tableau compare les temps de calculs des différentes méthodes. GARN, FARNA, NAST et ERNWIN ont été testés sur la même machine (Quad Core HT, 2.8 GHz Turbo, 10 MB, 1066MHz, 8 Go de RAM). iFoldRNA, RNAComposer et SimRNA ont été testés sur les serveurs dédiés. RNAComposer se connecte aux bases de données RNA FRABASE et RCSB (base de données PDB) pour reconstruire la molécule avec les connaissances qu'il possède déjà. Nous n'avons pas de connaissances sur les temps de calcul de RNAJAG. Les temps de calculs ont été réalisés sur 50 structures possibles. Pour RNAComposer, le temps de calcul est pour 10 structures (le serveur ne proposant pas plus de structures). Nous ne fournissons pas d'interactions tertiaires à NAST, qui n'opère alors pas d'autres repliements que les interactions de structures secondaires. Cela lui permet d'être plus rapide. Les paramètres utilisés dans les différentes approches sont les paramètres de base des logiciels.

Si nous visualisons la meilleure solution pour plusieurs approches (voir la figure 6.15), nous observons que GARN est la méthode qui atteint le repliement le plus proche de la structure PDB par rapport aux méthodes à grain fin sur les molécules de plus de 100 nucléotides.

Si nous ne considérons pas RNAComposer, alors nous observons que les structures de GARN sont en moyenne les meilleures structures, avec des RMSD plus faibles que les autres approches. Pour les molécules supérieures à 100 nucléotides, GARN est l'approche qui propose les RMSD minimales les plus faibles.

La table 6.2 compare le temps de calcul de 50 structures pour plusieurs molécules de notre *ensemble de test élargi*. Le temps de calcul pour GARN est inférieur à des approches comparables (comme FARNA ou ERNWIN). D'autres approches sont plus rapides, comme NAST (qui n'opère pas de repliement concernant les interactions tertiaires) ou encore RNAComposer (qui récupère les informations sur les bases de données).

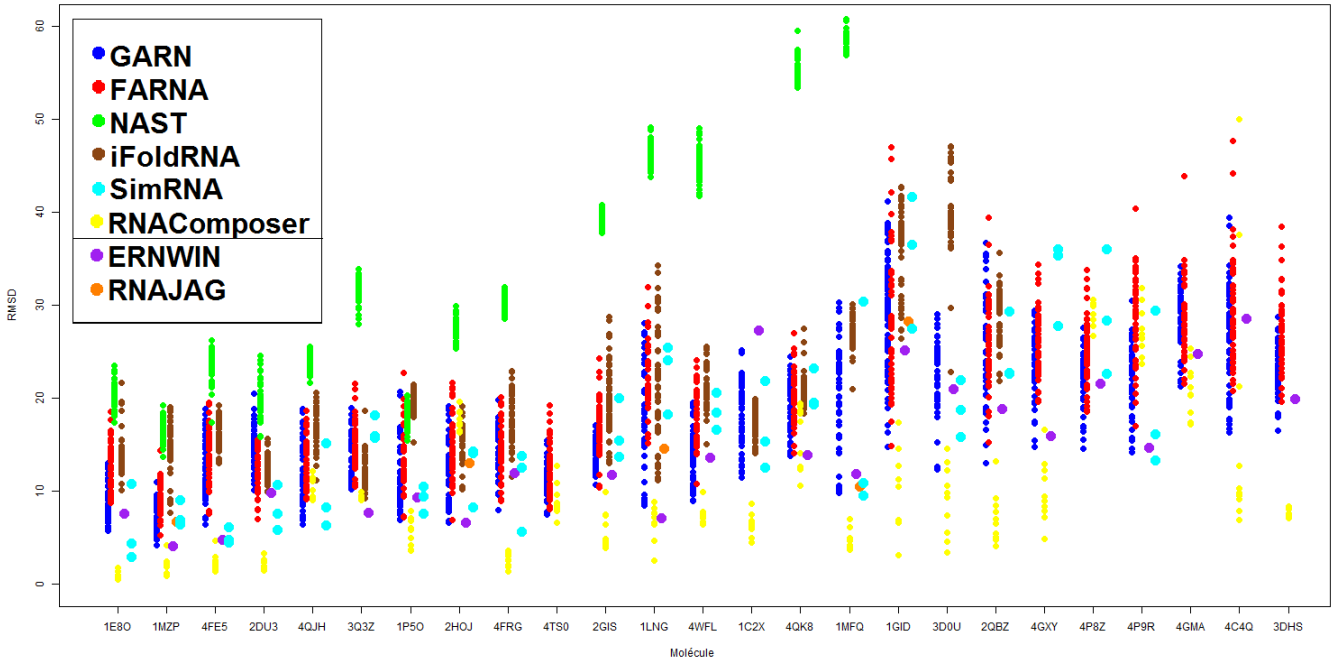


FIGURE 6.16 – **Visualisation de l'espace de solutions pour toutes les méthodes sur notre *ensemble de test élargi*.** Certaines méthodes ne peuvent pas générer de solutions pour tout l'*ensemble de test élargi*. Les résultats de RNAJAG proviennent de [Laing *et al.*, 2013, Kim *et al.*, 2014]. NAST peut replier toutes les molécules de notre *ensemble de test élargi* mais leurs RMSD ne sont pas représentées pour les grandes molécules (les RMSD dépassant les 80 Å), car NAST ne peut pas bien les replier sans interactions tertiaires.

6.3.4 Tests sur de grosses molécules

Dans ce qui précède, nous avons testé GARN sur des molécules de tailles différentes, allant de 50 à 268 nucléotides. Nous tentons ici notre méthode sur de très grosses molécules, de plus de 1000 nucléotides (voir la table 6.3 pour les molécules choisies). Si nous utilisons le même nombre d'itérations pour ces molécules que pour l'*ensemble de test élargi*, le calcul des solutions avec GARN dure très longtemps (plus de 48 heures). Nous avons donc adapté le nombre d'itérations de ces molécules pour obtenir 50 structures en 15 heures de calcul environ. Le nombre d'itérations de GARN est de $500 * NombredeJoueurs$. Nous réduisons le nombre d'itérations en effectuant ici $100 * NombredeJoueurs$. Nous n'avons pas pu calculer de structures avec les autres approches (les molécules possédant un nombre de nucléotides trop élevé).

ID PDB	Description	# Nucl.	# Joueurs	RMSD		
				Min	Moyenne	Max
1I97	16S RRNA	1514	293	43.93	57.66	79.12
1C2W	23S Ribosomal RNA	2904	505	62.87	76.59	80.25

TABLE 6.3 – **Résultat de l'ensemble de test des grosses molécules.** L'*ensemble de test des grosses molécules* contient deux molécules. Le tableau présente pour chaque molécule les RMSD pour 50 structures calculées avec notre approche GARN.

Les RMSD observés sont très importantes par rapport à ceux trouvés pour *ensemble de test élargi*. Les molécules étudiées ici possèdent néanmoins beaucoup plus de nucléotides. En effet, la molécule la plus grande de notre *ensemble de test élargi* est 3DHS, avec 268 nucléotides et une RMSD minimale de 16.54 Å. De nombreuses petites dissimilarités peuvent augmenter fortement la RMSD alors que les structures sont globalement similaires. Ici, la molécule 1I97 possède 1514 nucléotides (environ cinq fois plus que 3DHS), et GARN arrive à calculer une structure possédant une RMSD de 43.93 Å avec la structure PDB (environ 2.5 fois plus que pour 3DHS).

Il est difficile de visualiser des molécules aussi grosses, mais GARN permet tout de même de proposer des structures repliées pour de très grosses molécules.

6.4 Conclusion

Dans ce chapitre, nous avons développé une méthode de repliement pour toutes les structures d'ARN. Les structures secondaires peuvent être de tailles et de formes différentes. La première difficulté a été de modéliser cette structure quelque soit la forme de la molécule d'ARN, sans modifier fondamentalement le jeu.

Nous avons testé plusieurs paramètres possibles pour un repliement à gros grain. L'une des autres difficultés surmontées a été de trouver des paramètres s'accordant bien ensemble pour le repliement. Un paramètre qui permet un bon repliement sur une molécule peut, d'après d'autres paramètres, ne plus être un paramètre intéressant. Il est alors important d'observer les paramètres un à un en gardant une vision globale de leurs impacts.

Les paramètres liés à la discrétisation de l'espace nous ont permis de représenter statistiquement un modèle à gros grain d'une molécule. En restant dans une telle représentation statistique et globale, nous pouvons observer un repliement satisfaisant.

En effet, si nous observons les structures calculées par GARN et celles des principales approches existantes, GARN obtient de meilleures structures pour la majorité des grandes molécules (supérieures à 100 nucléotides). De plus, replier l'ARN grâce à une méthode à gros grains permet d'avoir une vision globale de la forme de la molécule, en un temps de calcul raisonnable (inférieur à l'heure pour des molécules possédant jusqu'à 200 nucléotides).

GARN permet donc de proposer un échantillonnage de structures possibles pour n'importe quelles formes et tailles d'ARN, avec des structures proches de celles recherchées.

7 Tri des structures obtenues

Dans le chapitre précédent, nous avons proposé une méthode permettant de trouver plusieurs structures possibles pour des molécules d'ARN. Afin d'étudier les structures calculées, il est utile de les trier. Un tri permet de ressortir une ou plusieurs structures potentiellement intéressantes, comme les structures proches de celle recherchée, sachant que nous ne connaissons pas à l'avance celle que nous souhaitons.

Dans la première section de ce chapitre, nous présenterons les critères de tri que nous allons tester. Ces critères dépendront soit du gain des joueurs introduit dans le chapitre précédent, soit de la compacité des structures. Dans la deuxième section, nous testerons ces critères pour choisir le plus pertinent. Nous étudierons dans la troisième section les résultats de notre tri sur les structures trouvées par notre méthode, GARN, et sur les structures trouvées par les autres approches actuelles.

7.1 Critères de tri

Après l'échantillonnage de l'espace de solutions des structures, nous avons plusieurs structures possibles sans connaître par avance la meilleure. La meilleure structure est celle dont la RMSD (voir la section 4.6, page 54) avec la structure recherchée (provenant de la *Protein Data Bank*) est la plus faible. Pour rechercher cette meilleure structure, nous allons trier les structures créées par notre méthode (présenté au chapitre précédent) d'après un critère de tri.

Pour cela, nous allons tester trois critères de tri :
- **Critère de gain total**, en calculant le gain total de la molécule, correspondant à la somme des gains des joueurs de la structure ;
- **Critère de gain minimum**, en récupérant le gain minimum parmi tous les joueurs de la structure ;
- **Critère de distance**, en calculant la distance entre les deux nœuds les plus éloignés de notre structure.

Les deux premiers critères se basent sur le gain des joueurs du chapitre précédent (voir la section 6.1.4, page 81). Durant le repliement (i.e. le jeu), les joueurs tentent de maximiser leur gain propre. Le gain permet à la structure de se replier en cherchant à atteindre des distances optimales entre les joueurs en fonction de leur type. Il est calculé pour chaque joueur. Ainsi le gain d'un joueur est une mesure locale. Nous pouvons supposer que plus les joueurs ont un gain élevé, plus le repliement est intéressant pour eux donc plus la structure est proche de celle recherché.

Pour le premier critère, le gain total, nous calculons la somme des gains de tous les joueurs pour attribuer un gain à la structure. Nous pouvons supposer qu'un gain élevé pour la structure (donc des gains en moyenne élevés pour chacun des joueurs) signifiera une RMSD faible (donc une structure proche de la structure PDB recherchée).

Le second critère, le gain minimum, consiste à récupérer les gains de chaque joueur et d'étudier le gain minimum parmi ces gains. Nous étudions ici la possibilité que le joueur qui a le moins maximiser son gain (donc qui a le gain le plus petit) se retrouve dans une situation plus ou moins défavorable d'après la structure. Si ce gain minimum est plus élevé sur une structure qu'une autre, alors ce gain pourrait indiquer que le repliement est mieux effectué et pénalise moins les joueurs.

Le troisième critère, la distance, se base sur la distance spatiale maximale au sein de la structure. Nous calculons la distance spatiale entre les deux nœuds les plus éloignés. Une distance plus faible que les autres indique une structure plus compacte. Cette distance nous permettra de trier les molécules d'après leur compacité.

Nous allons étudier ces trois critères pour connaître celui qui pourra au mieux trier nos résultats.

7.2 Recherche du critère le plus pertinent

Pour tester les différents critères, nous allons récupérer les structures proposées par notre méthode du chapitre précédent sur notre *ensemble de référence* (voir la table en annexe AT.1). Pour les molécules de cet ensemble, notre méthode propose 50 structures. Nous observons ensuite la distribution des RMSD (entre nos structures et la structure *PDB*) par rapport à nos différents critères. Nous présentons ici nos résultats en prenant en exemple quelques molécules de notre *ensemble de référence*. Nous pourrons ainsi observer s'il existe une corrélation entre la RMSD et nos critères. Si une corrélation existe alors nous pourrons observer une régression linéaire diagonale entre les deux variables (RMSD et critère).

7.2.1 Critère de gain total

La figure 7.1 permet de voir le rapport entre la RMSD et le critère de gain total pour les structures de quelques molécules de notre *ensemble de référence*. Le critère utilisé ici est la somme des gains de tous les joueurs pour une structure. Une somme élevée peut indiquer que tous les joueurs ont optimisé au maximum leur position, ou au contraire qu'un joueur a trouvé une position très optimale pour lui mais que les autres joueurs n'ont pu se replier correctement. Dans ce dernier cas, une somme élevée n'indique pas obligatoirement un repliement pertinent. Même si pour certaines molécules (comme 1KD5) nous pouvons voir une corrélation entre les deux, il nous est impossible de trier toutes les molécules d'après ce critère. Ce critère n'est donc pas judicieux pour trier nos résultats.

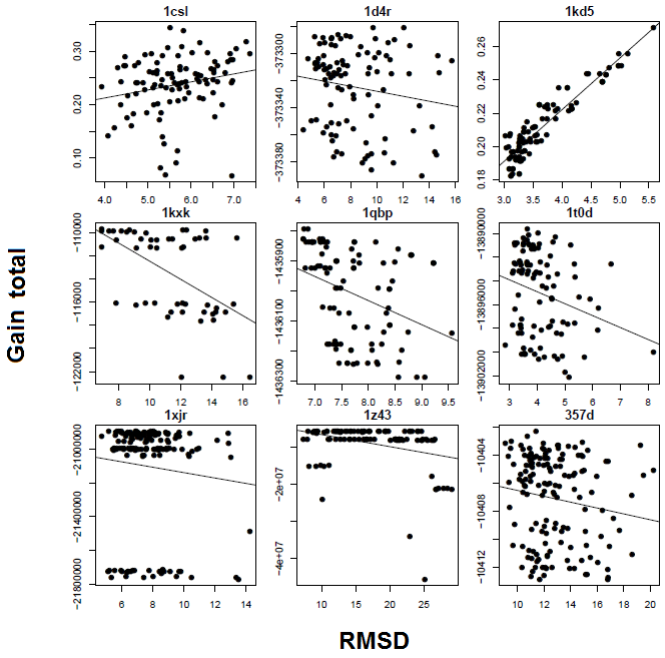


FIGURE 7.1 – **Rapport entre les RMSD et les gains totaux des structures.** Les figures représentent le rapport entre les RMSD trouvées (en abscisse) et les gains totaux de la structure (en ordonnée), pour plusieurs molécules de notre *ensemble de référence*. La ligne représente la régression linéaire de nos variables. La majorité des résultats ne montre pas de corrélation entre la RMSD et le gain de la structure.

7.2.2 Critère de gain minimum

La figure 7.2 permet de voir le rapport entre la RMSD et le critère de gain minimum pour les structures de quelques molécules de notre *ensemble de référence*. Le critère

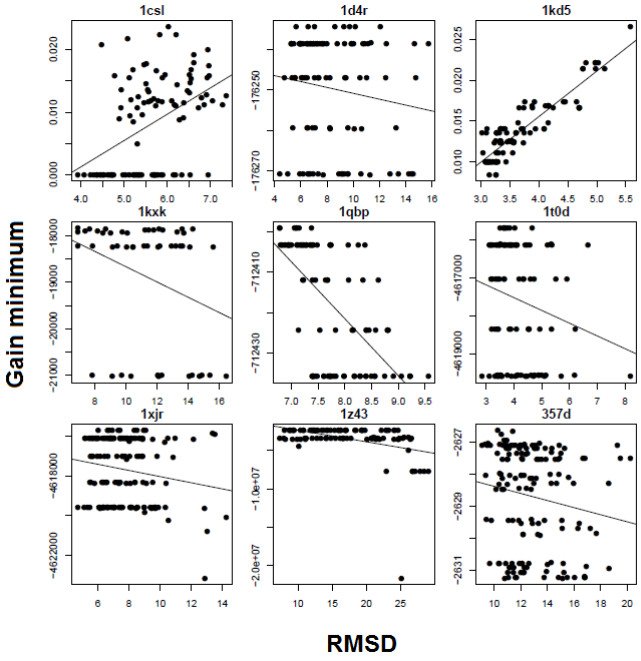


FIGURE 7.2 – **Rapport entre les RMSD et les gains minimaux des structures.** Les figures représentent le rapport entre les RMSD trouvées (en abscisse) et les gains minimaux de la structure (en ordonnée), pour plusieurs molécules de notre *ensemble de référence*. La ligne représente la régression linéaire de nos variables. La majorité des résultats ne montre pas de corrélation entre la RMSD et le gain de la molécule.

utilisé ici est le gain minimum des joueurs d'une structure. Un minimum élevé indique des gains globalement élevés pour les joueurs, et donc un bon repliement. Comme pour le précédent critère, certaines molécules comme 1KD5 montre une corrélation entre le critère et la RMSD. Le gain minimum peut être exactement le même entre plusieurs structures différentes. Par exemple, dans le cas de 1CSL, le gain minimum pour plusieurs structures est de 0. Il n'est donc pas possible de dissocier toutes les structures de gain minimum 0.

D'après les résultats, nous décidons que ce critère n'est donc pas intéressant.

7.2.3 Critère de distance

La figure 7.3 montre le rapport entre la RMSD et le critère de distance pour les structures de quelques molécules de notre *ensemble de référence*. Le critère utilisé ici est la distance entre les deux nœuds les plus éloignés d'une structure. Pour toutes les molécules, nous observons une régression linéaire diagonale, montrant une corrélation importante entre la

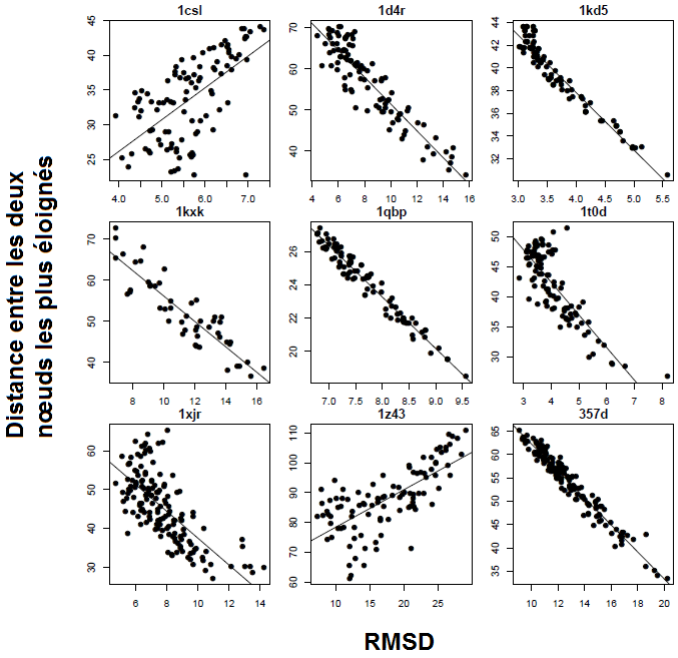


FIGURE 7.3 – **Rapport entre les RMSD et les distances entre les deux nœuds les plus éloignés des structures.** Les figures représentent le rapport entre les RMSD trouvées (en abscisse) et la distance entre les deux nœuds les plus éloignés de la structure (en ordonnée), pour plusieurs molécules de notre *ensemble de référence*. La ligne représente la régression linéaire de nos variables. Cette ligne diagonale permet de discerner un rapport entre la RMSD et la distance.

distance et la RMSD. Nous notons aussi que certaines molécules préfèrent une distance faible (comme 1CLS ou 1Z43), alors que d'autres molécules préfèrent une distance élevée (comme 1QBP ou 357D). Une distance faible indique une forte compacité, alors qu'une distance élevée indique une faible compacité par rapport aux autres structures. Certaines molécules préféreront ainsi une forte compacité (comme 1CLS), alors que d'autres préféreront l'inverse (comme 1QBP). Dans tous les cas, nous pouvons observer une corrélation (dans un sens ou l'autre) entre la RMSD et la distance. Ce critère est donc intéressant pour notre tri.

7.2.4 Critère choisi

Nous avons pu observer que les critères liés au gain n'étaient pas pertinents. Une somme des gains élevés peut signifier que de nombreux joueurs ont choisi une stratégie leur rapportant un gain moyennement élevé, ou qu'un seul joueur a un gain très élevé. Dans ces conditions, la somme n'est pas pertinente. Pour le gain minimum, il est souvent le

même pour plusieurs structures très différentes, ce qui ne permet pas de les différencier pour les trier. Le critère de distance, calculant la distance maximale entre deux nœuds au sein d'une structure, donne de bons résultats. Il permet de trier d'après la compacité de la structure, et ainsi d'indiquer si la structure de la molécule s'est fortement repliée ou non. Nous avons aussi observé que cette distance (entre les nœuds) permet de trouver de bonnes structures en la maximisant ou en la minimisant (par rapport aux autres distances maximales). Si nous minimisons cette distance alors la structure sera compacte. Au contraire, si nous maximisons cette distance alors la structure sera plus étendue (i.e. moins compacte).

Par conséquent, pour trier nos molécules, nous utiliserons le critère de distance. Nous choisirons alors la structure avec la distance la plus faible ainsi que la structure avec la distance la plus élevée. Nous proposons ainsi deux structures possibles avec des compacités différentes. Nous supposons que l'une de ces deux structures sera la meilleure ou approchant la meilleure.

7.3 Résultats de notre tri

Pour vérifier si le critère choisi grâce à notre étude sur notre *ensemble de référence* est pertinent, nous allons appliquer notre tri sur notre *ensemble de test élargi* (voir la table en annexe AT.7). Nous utiliserons le critère de distance pour le tri des structures.

Nous souhaitons, grâce à ce tri, obtenir la meilleure structure, ou au moins une structure proche de celle attendue. Pour cela, nous regarderons si l'une des deux structures trouvées grâce au tri (distance minimale ou maximale) se trouvent dans le 1^{er} quartile de notre échantillonnage. Par conséquent, sur 50 structures, nous souhaitons que celles proposées soient dans les 12 meilleures.

Dans la figure en annexe AF.4, nous observons que la régression linéaire des valeurs (RMSD et distance) est diagonale pour la majorité des molécules. Nous pouvons en conclure que le tri par distance peut donner de bons résultats sur les structures de nos molécules de l'*ensemble de test élargi*, permettant de corréler la distance et la RMSD. Nous comparons dans la table 7.1 les RMSD de ces structures avec le 1^{er} quartile de l'échantillonnage pour voir si notre tri nous permet de trouver une structure au moins dans le premier quartile (voir la figure en annexe AF.5 pour une comparaison avec l'ensemble de l'échantillonage). Nous observons que nous arrivons dans presque tous les cas à trouver une structure présente dans le premier 1^{er} quartile de notre échantillonnage, voir la meilleure structure. Seule la molécule 4WFL n'a pas pu être triée correctement. La structure de cette molécule, composée de deux 3-jonctions, possède une jonction compacte et une jonction plus étendue. Prendre les distances minimales et maximales ne nous permet pas de trouver une très bonne structure. Pour les molécules comme 4FE5, 4QJH ou encore 4C4Q, nous arrivons grâce à ce tri à trouver la structure avec la RMSD

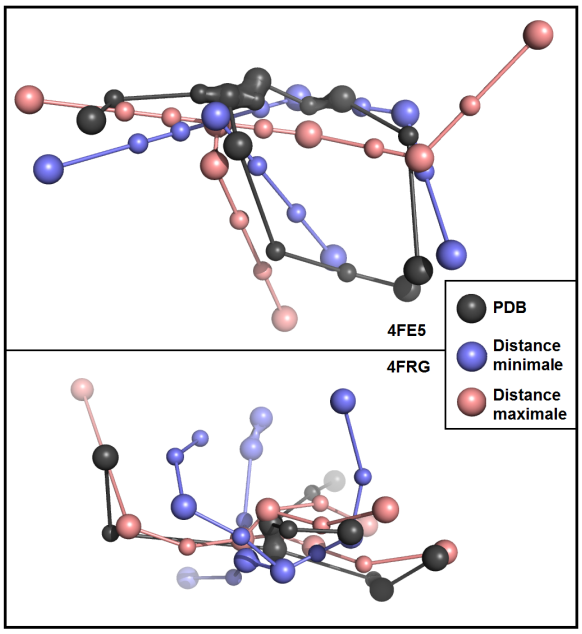


FIGURE 7.4 – **Visualisation des structures choisies d'après la distance entre les deux nœuds les plus éloignés.** Le tri permet de sortir de l'échantillonnage deux structures : celle avec la distance minimale (en bleu) ou maximale (en rose) entre les deux nœuds les plus éloignés de la structure. Nous superposons la structure PDB (en noir). Les structures de la molécule 4FE5 sont mieux triées en prenant la distance minimale alors que les structures de la molécule 4FRG se trient mieux avec une distance maximale.

la plus faible de tout l'échantillonnage.

La figure 7.4 représente les structures de distances minimales et maximales pour deux molécules (4FE5 et 4FRG). Nous observons qu'il y a une différence de repliement, et donc de compacité, entre les deux structures choisies par le tri. En effet, la molécule 4FE5 préfère une compacité plus forte que la molécule 4FRG.

Notre critère de tri choisi nous permet de replier correctement notre *ensemble de test élargi*. Nous proposons donc grâce à ce tri deux structures possibles pour une molécule.

7.4 Tri des approches existantes de l'état de l'art

Nous avons testé notre méthode de tri sur les résultats de notre méthode GARN. Nous allons maintenant tester notre tri sur les approches existantes dont nous avons les structures et que nous pouvons plonger dans notre modèle à gros grain (FARNA, NAST,

PDB ID	Nucl.	1^{er} Quartile	RMSD Dist. Min	RMSD Dist. Max
1E8O	50	7.41	7.31	14
1MZP	55	5.95	5.71	9.24
4FE5	68	10.03	**6.43**	17.59
2DU3	71	13.72	12.47	18.81
4QJH	74	9.12	12.45	**6.44**
3Q3Z	77	11.51	11.23	18.73
1P5O	77	8.70	17.82	8.51
2HOJ	79	9.43	8.88	16.07
4FRG	84	12.80	14	**7.99**
4TS0	89	10.39	20.35	9.33
2GIS	94	12.62	16.39	11.94
1LNG	97	15.30	13.83	31.4
4WFL	107	11.69	12.58	18.49
1C2X	120	15.33	13.12	17.61
4QK8	124	17.14	16.81	23.94
1MFQ	127	17.33	16.51	27.69
1GID	158	26.04	22.88	37.72
3D0U	161	20.09	15.26	26.86
2QBZ	161	22.14	20.58	34.23
4GXY	172	21.08	16.3	30
4P8Z	188	20.01	18.88	28.58
4P9R	192	20.49	19.65	29.52
4GMA	210	27.22	23.92	33.22
4C4Q	233	24.68	37.52	**16.32**
3DHS	268	21.51	24.26	22.84

TABLE 7.1 – **Résultats du tri par distance sur notre *ensemble de test élargi* pour GARN.** Ce tableau compare la RMSD pour la structure ayant une distance minimale (Min) ou maximale (Max) entre les deux points les plus éloignés, et la limite du 1^{er} quartile pour les RMSD des structures. Nous indiquons en bleu les valeurs se trouvant sous la barre du 1^{er} quartile. Nous notons en gras la valeur si celle-ci est la valeur de RMSD minimale de l'échantillonnage. Pour presque toutes les molécules, nous trouvons une structure dans le 1^{er} quartile.

iFoldRNA et RNAComposer). Nous ne testons pas sur les méthodes à gros grains, ne pouvant pas passer leur modèle dans le notre. Nous ne testons pas aussi sur SimRNA, le serveur ne renvoyant que trois structures. Nous trions les structures de ces approches pour en sortir deux structures. Les résultats sont disponibles dans la table annexe AT.9.

Nous ne proposons pas à NAST d'interactions tertiaires. Les structures proposées sont donc peu repliées. Dans ce cas-là, la distance doit être minimale entre les deux nœuds les

plus éloignés pour trouver la structure la plus repliée.

FARNA et iFoldRNA proposent des échantillons assez larges de structures plus ou moins compactes pour les molécules. Pour un grand nombre de molécules, notre tri propose une solution avec une RMSD inférieure au 1^{er} quartile. Nous remarquons que le choix entre une distance minimale et maximale varie entre les différentes approches. Par exemple, pour la molécule 2DU3, le tri de GARN donnera la meilleure structure avec une distance minimale alors que le tri de FARNA se fera mieux avec une distance maximale (voir la figure 7.5).

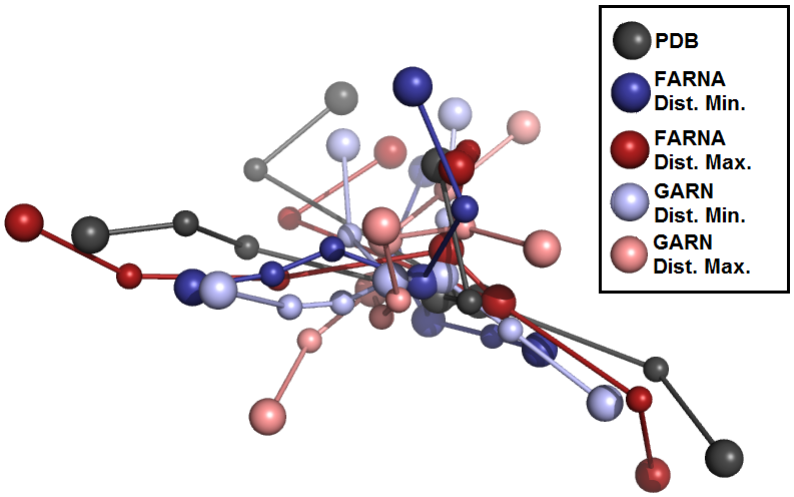


FIGURE 7.5 – **Visualisation des deux structures triées de 2DU3 pour GARN et FARNA.** Le tri permet de sortir de l'échantillonnage deux structures : celle avec la distance minimale (en bleu) ou maximale (en rose) entre les nœuds les plus éloignés de la structure. Nous superposons la structure PDB (en noir). La molécule 2DU3 est mieux triée en prenant la distance maximale pour FARNA et la distance minimale pour GARN.

Inversement, le tri des structures pour 2GIS donnera de meilleurs résultats avec une distance maximale pour GARN et une distance minimale pour iFoldRNA (voir la figure 7.6). Les échantillons des approches n'offrent pas toutes les mêmes structures. Sur une molécule compacte, une approche pourrait proposer un échantillon de structures trop compactes et la structure avec une distance maximale sera alors la meilleure. Inversement, si l'échantillon n'est pas assez compact, la distance minimale pourra être préférée.

RNAComposer propose 10 structures, nous travaillons donc sur un petit échantillon où les structures sont très proches. Par exemple, les RMSD des structures pour la molécule 3Q3Z sont comprises entre 9.05 Å et 9.95 Å. Les échantillons très restreints ne nous

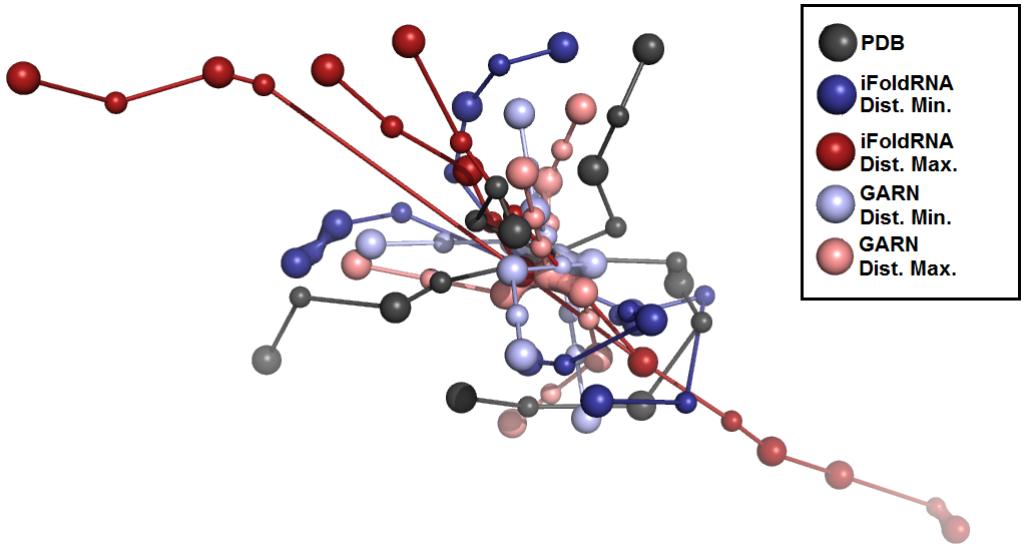


FIGURE 7.6 – **Visualisation des deux structures triées de 2GIS pour GARN et iFoldRNA.** Le tri permet de sortir de l'échantillonnage deux structures : celle avec la distance minimale (en bleu) ou maximale (en rose) entre les nœuds les plus éloignés de la structure. Nous superposons la structure PDB (en noir). La molécule 2GIS est mieux triée en prenant la distance minimale pour iFoldRNA et la distance maximale pour GARN.

permettent pas d'avoir un bon tri, notre différenciation entre deux structures n'étant pas suffisante. Les molécules avec un plus large échantillonnage (comme 3D0U ou 4GXY, avec des différences de RMSD de 10 Å) sont mieux triées par notre méthode, et nous pouvons trouver une structure de faible RMSD.

Il est difficile de choisir entre la distance minimale ou maximale pour choisir la meilleure structure. En effet, ce choix dépend de la compacité de la structure et aussi de la largeur de l'échantillonnage. Ne sachant pas à l'avance la compacité recherchée, notamment au sein d'un échantillonnage, nous ne pouvons choisir entre les deux structures.

7.5 Conclusion

Les approches actuelles ainsi que GARN proposent plusieurs structures possibles pour les molécules étudiées. Certaines approches proposent un tri d'après les potentiels calculés durant leurs repliements.

Pour trier nos solutions, nous avons introduit plusieurs critères de tri, liés au gain de

notre jeu ou à une mesure de compacité. Nous avons observé que les critères liés au gain n'étaient pas pertinents. La mesure de compacité, correspondant à la distance entre les deux nœuds les plus éloignés de la structure, est un critère de tri pertinent. Ce critère nous propose de choisir entre une compacité importante ou non de notre structure. Ne connaissant pas à l'avance la compacité recherchée, notre tri propose deux structures : une avec la distance calculée minimisée et une maximisant cette distance.

Après avoir développé une méthode nous permettant de sortir plusieurs structures possibles (par exemple une cinquantaine), nous avons proposé ici de les trier pour pouvoir proposer, au final, deux structures possibles pour la molécule, dont l'une possédant une RMSD faible pour notre échantillonnage.

Conclusion Partie IV

8 Conclusion

Au cours de cette thèse, nous avons développé une nouvelle méthode de prédiction de structure tridimensionnelle d'ARN. Cette méthode utilise une modélisation à gros grain de la molécule, couplée avec des algorithmes de minimisation de regret. Notre paradigme de base est qu'une structure stable d'une molécule d'ARN peut être vue comme un équilibre de Nash (en théorie des jeux) ou comme un jeu où les éléments de la molécule souhaitent minimiser leur regret. Pour cela, nous avons modélisé notre problème comme un jeu, afin d'appliquer des algorithmes de minimisation de regret pour nous permettre d'approcher des équilibres de Nash.

Le modèle de notre jeu est donc basé sur la représentation à gros grain de notre molécule. Cette représentation consiste à voir chaque élément d'une structure secondaire (structure en 2D) comme les nœuds d'un graphe. Ces nœuds sont les joueurs de notre jeu de repliement. Nous assignons ensuite à chaque joueur un ensemble de stratégies correspondant à des directions dans l'espace, permettant différents repliements tridimensionnels. Pour finaliser notre jeu, nous avons calculé des potentiels correspondant aux distances observées entre chaque élément de la structure secondaire des molécules. Les potentiels ont été réalisés en utilisant des statistiques sur des structures connues de molécules. Ces potentiels sont utilisés pour calculer les gains des joueurs. Ce jeu peut être paramétré de plusieurs façons différentes afin d'optimiser le repliement. Nous avons détaillé et testé dans ce document plusieurs paramètres possibles : la distance entre les joueurs, la possibilité d'empêcher le repliement d'un joueur (i.e. la rigidité), l'utilisation de différents algorithmes ou méthodes de calcul, etc.
Ce jeu nous permet de trouver plusieurs structures possibles. Nous avons ainsi un large échantillonnage de structures pour une molécule donnée. Notre méthode, nommée GARN, calcule des structures plus proches des structures réelles pour un temps de calcul égal ou plus faible que les approches communément utilisées. Enfin, il est possible de proposer deux structures avec des compacités très différentes, dont l'une des structures est proche de celle recherchée. GARN a donc atteint les objectifs que nous nous étions donnés : réussir à replier rapidement tous les types d'ARN pour avoir une vision globale de la

forme de la molécule. GARN est téléchargeable librement sur (http://garn.lri.fr).

Pour arriver à finaliser cette méthode, nous avons procédé par étapes durant tout ce document.

Dans la première partie de nos travaux, nous avons établi un état de l'art des connaissances sur la structure de l'ARN, sur la théorie des jeux et sur la minimisation de regret.

Dans la deuxième partie de nos travaux, nous avons étudié des jeux préliminaires pour savoir si notre paradigme de base était juste. Nous avons recherché des équilibres de Nash pour des jeux simples de repliement d'une chaîne dans un espace en deux dimensions et en trois dimensions. Nous avons prouvé qu'une chaîne, plongée dans un espace en trois dimensions, peut se replier en un équilibre de Nash pur pour former des interactions. Il existe donc bien des équilibres correspondant à une structure maximisant ses interactions en minimisant ses repliements. Cette première analyse nous a conforté dans l'idée de tester notre idée de jeu pour le repliement.

Nous avons donc développé une première approche de notre méthode fonctionnant sur des structures d'ARN de tailles moyennes. Cette approche utilisait une grille 3D pour discrétiser l'espace et les solutions. Nous avons décrit dans cette partie les bases de notre modèle de jeu. Nous avons aussi testé plusieurs paramètres possibles, nous permettant de construire notre jeu préliminaire. Nous avons pu observer que des paramètres uniques pour toutes les molécules ne permettaient pas un repliement correct de l'ensemble des molécules. Nous avons alors proposé un regroupement des molécules d'après des informations sur leur structure secondaire pour le choix des paramètres. Nous avons aussi conclu, après comparaisons des résultats, que les algorithmes de minimisation de regret donne dans notre jeu de meilleurs résultats que les algorithmes de recherche d'équilibre ou que les algorithmes de Monte-Carlo. Sur ces bases, nous avons ensuite développé une méthode pouvant prendre en compte toutes les molécules d'ARN.

Dans la troisième partie, nous avons donc étendu notre jeu à toutes les molécules d'ARN tout en l'améliorant. Les paramètres que nous avons testés (par exemple la distance entre les éléments de l'ARN dépendant du nombre de nucléotides) nous ont permis de proposer une méthode de repliement qui s'adapte à chaque molécule. Pour cela, nous avons proposé une démarche permettant de choisir automatiquement les paramètres pour chaque molécule. Avec notre méthode, nous arrivons à calculer des structures de molécules plus proches des structures réelles que la majorité des approches de l'état de l'art. Nous avons aussi testé notre méthode sur de très grosses molécules (plus de 1500 nucléotides). GARN permet de calculer plusieurs structures pour ces très grosses molécules, mais nous n'avons pas pu nous comparer avec les logiciels existants (ces derniers ne pouvant pas calculer de structures pour des molécules aussi grosses). Nous avons pu aussi observer que, sur un modèle à gros grain, la recherche de la précision n'était pas la meilleure approche pour une amélioration du modèle. En effet, certains paramètres (comme la distance entre

les joueurs adjacents) fonctionnent mieux avec une représentation très discrète qu'avec une représentation plus continue. Nous perdons de l'information lors du passage à gros grain, mais nous gagnons des informations sur le repliement global. Par conséquent, la perte d'information lors du passage à gros grain s'équilibre avec la discrétisation pour le repliement à ce niveau de modélisation. Nous avons enfin proposé une méthode de tri pour les structures calculées par notre approche GARN. Après avoir testé des critères de tri liés aux potentiels de notre jeu, nous avons observé qu'un critère de distance entre les joueurs, représentant la compacité, était plus pertinent. Nous proposons ainsi une méthode pour proposer à l'utilisateur deux structures (l'une très compacte et l'autre peu compacte), dont l'une étant une des meilleures structures proposées.

En conclusion, nous avons réussi à développer une approche et un logiciel permettant de calculer des repliements proches des structures réelles pour des molécules d'ARN. Cette approche peut être améliorée et amener à d'autres études.

Dans ce document, plusieurs paramètres pour notre jeu ont été testés. Durant notre étude, nous avons tenté de nombreux autres paramètres que nous n'avons pas exploités car les premiers résultats n'étaient pas probants sur notre *ensemble de référence*. Plusieurs d'entre eux auront pu être améliorés pour être utilisés dans GARN. Nous avons testé, par exemple, une distance entre les joueurs dépendant de statistiques sur la largeur des 2-jonctions. Cette distance n'a pas donné de bons résultats dans son ensemble. Cela nous a surpris puisqu'elle se basait sur une analyse statistique plus fine que celle que nous avons faite lors de notre première étude des molécules. Nous avons aussi testé de nombreux autres paramètres, comme une diffusion des gains d'un joueur sur ces joueurs adjacents ou encore en modifiant le gain du joueur pour ne prendre que le potentiel maximum. L'objectif est d'affiner chaque paramètre, même ceux que nous avons rejetés au début, pour une optimisation des résultats et pour une possibilité d'évolution de GARN.

Nous avons notamment utilisé des potentiels pour calculer le gain des joueurs. Nos potentiels sont basés sur les distances entre les joueurs. La méthode ERNWIN, récemment développé, utilise un potentiel assez semblable, proposant aussi plusieurs autres potentiels d'après des statistiques sur d'autres informations que la distance. Dans cette optique, nous pourrions ajouter un potentiel sur les angles des joueurs, ou encore sur les nucléotides des joueurs. Il est aussi possible d'ajouter un potentiel représentant la forme globale de la molécule pour imposer aux joueurs une vision globale de la structure, en utilisant par exemple les critères vus dans notre tri.

Dans le chapitre 6, nous avons replié des molécules possédant plus de 1000 nucléotides, sans adapter les paramètres du jeu. Les RMSD trouvées sont élevées mais indique une possibilité de repliement via notre méthode. Il serait intéressant de pousser notre

étude pour mieux l'adapter à ces molécules. Le paramétrage du jeu pourrait s'adapter à différentes formes de molécules, en changeant notamment le potentiel. Il serait aussi possible de replier certaines parties de la molécule plutôt que de la replier globalement. En travaillant sur chaque partie d'une très grosse molécule, le repliement pourrait être plus précis localement et permettre un repliement global plus correct.

Dans le dernier chapitre, nous avons trié les structures calculées par GARN sans les regrouper. Ce tri nous a permis de proposer deux structures très différentes (très compacte et plus allongée). En utilisant des méthodes de regroupement, nous pourrions proposer plusieurs structures différentes. L'utilisation de méthode de partitionnement de données (*clustering*) serait une technique pour proposer quelques structures très différentes les unes des autres. Une piste est de vouloir regrouper les conformations en utilisant des mesures classiques comme la RMSD. Une autre façon est de les classifier en fonction de la forme (globuleuse ou allongée), comme lors de notre tri.

GARN est une méthode développée pour toutes les molécules d'ARN, mais pourrait être adaptée à des familles de molécules différentes. En effet, les paramètres de jeu choisis pourraient s'adapter à certaines familles d'ARN, ou à certaines tailles de molécules. Nous pouvons aussi modifier les paramètres pour s'adapter à d'autres types de molécules, comme les protéines, tant que nous leur trouvons une représentation à gros grain favorable à un repliement par jeu.

Il est aussi logique de vouloir redescendre à des niveaux de modélisation proches des nucléotides ou des atomes. Avec une représentation à gros grain, nous n'avons pas de visibilité au niveau du nucléotide ou de l'atome. Nous pourrions utiliser notre connaissance de la forme globale de la structure pour recréer une structure à grain fin. Par exemple, nous assemblerions plusieurs éléments aux niveaux atomiques sur les éléments de notre structure à gros grain. Nous pourrions aussi chercher un jeu permettant de placer les nucléotides (ou les atomes) d'après la forme globale de la molécule. Enfin, il serait possible d'intégrer notre structure à des logiciels pouvant créer une structure atomique de l'ARN.

Redescendre au niveau des nucléotides permettrait aussi une meilleure comparaison avec les solutions existantes. Il serait aussi intéressant de trouver un moyen de comparer deux modèles à gros grain (comme GARN et ERNWIN) malgré la différence de représentation. Une approche basée sur la représentation à grain plus fin est une possibilité.

Enfin, le repliement de l'ARN se fait grâce à plusieurs tours de jeu, induisant une évolution de la structure. Nous pourrions étudier chaque tour de jeu et l'évolution des stratégies testées pour en ressortir une dynamique de repliement des molécules d'ARN, et la comparer avec les connaissances actuelles de dynamique de repliement des molécules.

Pour calculer les structures possibles, nous avons utilisé des algorithmes de minimisation de regret avec des potentiels. La minimisation de regret peut être l'équivalent d'une descente gradient pour des fonctions convexes. Hors, nos potentiels (Lennard-Jones ou

Gauss) ne sont pas des fonctions convexes. Il serait intéressant d'analyser l'impact d'une fonction non-convexe sur la minimisation de regret et voir s'il est possible de trouver un équivalent en descente de gradient ou en recherche d'un minimum local.

Enfin, l'utilisation d'une méthode à gros grains telle que GARN peut permettre d'avoir une première idée de la forme réelle de la molécule. Cette vision peut être utilisée pour créer plus rapidement des molécules avec des formes voulues pour induire une fonction recherchée ou une possibilité de raccrochement à une autre molécule. Une création rapide de nouvelles molécules d'ARN peut permettre d'améliorer entre autres la conception de médicament. Il est donc toujours intéressant de continuer à développer des méthodes donnant une idée rapide de la forme d'une molécule pour faciliter tous les travaux concernant leur structure 3D.

Annexes Partie V

ID	Description	Famille	# Nucl.
157D	RNA duplex	Autre ARN	24
1CSL	HIV-1 rev binding site	ARN Viral	28
1D4R	Human SRP helix 6	SRP RNA	58
1DQF	5S rRNA	ARN Ribosomal	19
1DUH	4.5s RNA	SRP RNA	90
1DUQ	HIV-1 rev binding site	ARN Viral	26
1F1T	RNA duplex	Autre ARN	38
1F27	RNA pseudoknot	Autre ARN	30
1FIR	HIV reverse-transcription primer tRNA(Lys,3)	ARN de transfert	76
1G2J	RNA duplex containing phenyl-ribonucleotides	Autre ARN	16
1I9V	Phenylalanine tRNA	ARN de transfert	76
1I9X	snRNA	Autre ARN	26
1J9H	RNA nonamer	Autre ARN	18
1JZV	Bulged RNA from the SL2 stem-loop	ARN Viral	17
1K9W	HIV-1 RNA dimerization initiation site	ARN Viral	46
1KD5	RNA duplex	Autre ARN	22
1KFO	RNA duplex	Autre ARN	38
1KH6	RNA tertiary domain	ARN Viral	53
1KXK	Ai5gamma group II self-splicing intron	Autre ARN	70
1L2X	Viral (beet western yellow virus) rna pseudoknot	ARN Viral	28
1L3Z	RNA heptamer	Autre ARN	14
1MHK	26-nucleotide RNA containing a hook-turn	Autre ARN	52
1MME	Hammerhead ribozyme	Ribozyme	82
1MSY	Sarcin/Ricin domain from 23s RNA	ARN Ribosomal	27
1NBS	Ribonuclease P RNA	Ribozyme	155
1NUJ	Leadzyme	Ribozyme	24
1P79	Bulged RNA tetraplex	Autre ARN	20
1Q93	Sarcin/Ricin domain mutant from 28s RNA	ARN Ribosomal	27
1QBP	RNA duplex	Autre ARN	30
1SA9	RNA octamer	Autre ARN	16
1SDR	Shine-Dalgarno region of 16S rRNA	ARN Ribosomal	24
1T0D	Ribosomal decoding site	ARN Ribosomal	33
1T0E	Ribosomal decoding site	ARN Ribosomal	35
1U9S	Ribonuclease P RNA	Ribozyme	159
1X8W	Tetrahymena ribozyme	Ribozyme	247
1X9C	Hairpin Ribozyme	Ribozyme	61
1X9K	Hairpin Ribozyme	Ribozyme	62
1XJR	SARS virus genome stem-loop II motif (s2m)	ARN Viral	47
1Y0Q	Group I ribozyme	Ribozyme	248
1YFG	Yeast initiator tRNA	ARN de transfert	75
1YKQ	Diels-Alder Ribozyme	Ribozyme	49

ID	Description	Famille	# Nucl.
1YZD	RNA duplex	Autre ARN	32
1Z43	SRP19 RNA	Autre ARN	101
1Z58	23s ARN Ribosomal	ARN Ribosomal	2877
205D	RNA duplex	Autre ARN	24
255D	RNA duplex	Autre ARN	24
259D	RNA octamer	Autre ARN	16
280D	RNA duplex	Autre ARN	24
2A0P	RNA octamer	Autre ARN	16
2A2E	Ribonuclease P RNA	Ribozyme	334
2A64	Ribonuclease P RNA	Ribozyme	414
2AO5	RNA duplex	Autre ARN	20
2B8R	HIV-1 RNA dimerization initiation site	ARN Viral	46
2G32	RNA octamer	Autre ARN	16
2G3S	RNA octamer	Autre ARN	16
2G91	RNA nonamer	Autre ARN	17
2H0S	glmS ribozyme	Ribozyme	145
2IL9	Ribosomal Binding Domain of the IRES RNA	ARN Viral	282
2NOK	RNA internal ribosome entry site (IRES) domain	ARN Viral	44
2OE6	Ribosomal decoding A site	ARN Ribosomal	33
2TRA	Yeast aspartic acid tRNA	ARN de transfert	73
333D	RNA octamer	Autre ARN	16
353D	5S rRNA	ARN Ribosomal	23
357D	5S rRNA	ARN Ribosomal	61
361D	5S rRNA	ARN Ribosomal	20
377D	Alternating A-RNA hexamer	Autre ARN	12
387D	RNA pseudoknot with 3d domain swapping	Autre ARN	52
397D	HIV-1 trans-activation response region RNA stem	ARN Viral	27
402D	RNA octamer	Autre ARN	16
405D	RNA duplex	Autre ARN	32
406D	RNA duplex	Autre ARN	34
409D	RNA octamer	Autre ARN	16
413D	RNA duplex	Autre ARN	26
433D	RNA duplex	Autre ARN	28
434D	acceptor stem of tRNA(Ala) from Escherichia coli	ARN de transfert	14
438D	RNA nonamer	Autre ARN	18
472D	Octamer RNA with G.G/U.U tandem wobble base pairs	Autre ARN	16

TABLE AT.1 – **Ensemble de référence.** L'*ensemble de référence* contient 77 molécules. Les molécules de cet ensemble de référence proviennent de [Bernauer *et al.*, 2011].

PDB ID	Description	# Nucleotides	# de 3-jonctions
1E8O	7SL RNA	49	1
1MZP	Fragment of 23S rRNA	55	0
4FE5	Guanine riboswitch aptamer domain	67	1
4QJH	Twister Ribozyme	74	1
4TS0	Spinach RNA aptamer	89	0
1LNG	7S.S SRP RNA	97	1
4WFL	Bacterial SRP Alu domain	107	2
4QK8	C-di-AMP riboswitch	124	2
1MFQ	7S RNA of human SRP	127	1

TABLE AT.2 – **Ensemble de test.** L' *ensemble de test* contient 10 molécules de tailles différentes. Nous notons ici le nombre de 3-jonctions au sein de la molécule.

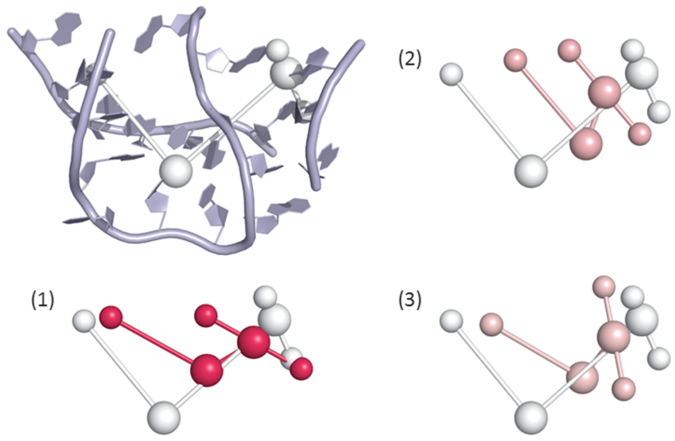


FIGURE AF.1 – **Structure de la 3-jonction pour la molécule 1MFQ.** Le panneau du haut montre la structure et son graphe GARN associé. Le panneau (1) montre la meilleure 3-jonction obtenu avec la méthode préliminaire de GARN (en rose) superposée sur la structure PDB (en blanc). Les panneaux (2) et (3) montre les secondes meilleures 3-jonctions (en rose) superposées à la structure PDB (en blanc).

ID	# Nucl.	# Joueurs	GARN Min	GARN Max	MC-Sym Min	MC-Sym Max	Farna Min	Farna Max	NAST Min	NAST Max
1SA9	16	3	3.97	3.97	0.67	3.31	**0.24**	4.22	0.34	2.24
333D	16	3	3.81	3.81	**0.34**	3.57	0.48	3.3	0.36	2.48
402D	16	3	4.15	4.15	**0.14**	4.7	0.29	3.81	NA	NA
409D	16	3	3.93	3.93	0.49	2.95	**0.31**	4.09	0.56	2.30
1JZV	17	3	4.3	9.24	**0.15**	3.94	0.82	6.11	2.51	4.86
1J9H	18	5	4.56	11.53	**1.14**	4.2	1.76	6.77	2.84	4.91
1DQF	19	4	5.39	5.39	**0.84**	3.49	1.46	8.58	NA	NA
1KD5	22	5	9.54	9.54	NA	NA	**1.56**	12.68	1.81	5.71
205D	24	3	4.79	4.79	**0.22**	4.62	0.24	5.1	0.26	2.70
255D	24	3	5.4	5.4	0.66	3.86	**0.21**	6.07	0.32	3.14
280D	24	3	4.36	4.36	**0.22**	5.22	0.28	6.09	**0.22**	2.10
157D	24	5	2.17	12.94	**0.66**	3.31	0.77	7.66	1.11	4.68
1NUJ	24	5	7.07	7.07	NA	NA	**1.62**	10.48	2.07	6.01
1DUQ	26	5	4.86	6.16	**0.66**	7.24	2.41	9.03	NA	NA
1I9X	26	5	6.74	8.35	**1.82**	5.52	3.26	9.25	4.37	9.64
387D	26	5	7.69	7.69	NA	NA	3.95	9.96	**3.63**	7.88
413D	26	5	8.44	8.44	NA	NA	**1.17**	13.29	4.23	6.62
1Q93	27	4	7.66	7.66	NA	NA	0.98	9.64	**0.90**	5.85
1MSY	27	5	4.71	10.06	NA	NA	**1.27**	8.3	1.58	5.54
397D	27	6	4.09	10.16	NA	NA	**2.32**	8.73	3.15	6.04
1QBP	28	5	4.2	8.98	**1.02**	6.35	1.99	8.3	2.07	4.97
1CSL	28	7	3.86	9.69	NA	NA	**2.96**	11.99	5.86	8.84
405D	32	5	2.43	4.73	**0.62**	2.64	1.18	8.5	3.42	6.63
1T0D	33	8	**3.14**	13.87	NA	NA	NA	NA	3.67	7.06
2OE6	33	8	4.77	14.12	NA	NA	NA	NA	**4.67**	7.33
406D	34	5	3.47	9.7	NA	NA	NA	NA	**3.13**	4.87
1T0E	35	9	**3.53**	14.66	NA	NA	NA	NA	3.97	6.49
1F1T	38	7	3.68	9.37	**2.17**	6.86	NA	NA	4.17	7.48
1KFO	38	11	6.39	14.91	NA	NA	NA	NA	**5.99**	10.72
1MME	41	8	**5.76**	10.44	NA	NA	NA	NA	NA	NA
2NOK	44	12	**4.88**	16.77	NA	NA	NA	NA	6.48	10.65
1XJR	47	12	6.23	13.16	5.48	10.82	NA	NA	**5.44**	9.33
1D4R	57	10	8.07	21.61	NA	NA	NA	NA	**4.17**	9.03
357D	60	8	12.17	25.42	5.27	23.23	NA	NA	**3.28**	10.85
1KXK	70	10	8.35	21.28	**5.06**	12.07	NA	NA	6.80	14.00
1DUH	90	13	**8.63**	33.33	NA	NA	NA	NA	NA	NA
1Z43	101	18	**8.98**	15.54	NA	NA	NA	NA	31.24	55.70

TABLE AT.3 – **Comparaison des RMSD pour l'*ensemble de référence*.** Ce tableau montre les RMSD de certaines molécules de l'*ensemble de référence* pour notre approche préliminaire (GARN) et pour d'autres approches de l'état de l'art. La RMSD minimale pour chaque molécule est indiquée en bleu. Toutes les molécules de l'*ensemble de référence* ne peuvent être repliées par notre méthode préliminaire (en fonction de leur structure secondaire). Ce tableau montre que les RMSD des structures calculées par notre approche préliminaire peuvent être très faibles (inférieures à 5 Å). Nous notons aussi que pour les petites molécules (inférieures à 30 nucléotides), les approches à grain fin proposent des RMSD plus faibles que notre approche.

ID	# Nucl.	# Joueurs	Ratio H/J	RMSD	Famille 1	Famille 2	Famille 3
1MZP	55	8	1.3	min	4.32	6.84	5.43
				max	10.58	11.62	11.85
# de solutions avec une RMSD < 5Å (sur 50 solutions)					7	1	3
4TS0	89	21	1.28	min	10.42	11.98	10.53
				max	23.13	22.50	18.31
# de solutions avec une RMSD < 5Å (sur 50 solutions)					4	1	2
1E8O	49	8	2	min	8.64	6.82	8.75
				max	16.2	15.40	12
# de solutions avec une RMSD < 8Å (sur 50 solutions)					0	13	0
4FE5	67	14	1.75	min	8.21	7.14	10.56
				max	18.27	14.53	13.73
# de solutions avec une RMSD < 9Å (sur 50 solutions)					3	2	2
4QJH	72	15	1.4	min	10.13	7.87	10.56
				max	20.32	14.93	15.31
# de solutions avec une RMSD < 9Å (sur 50 solutions)					0	3	0
1LNG	97	16	1.6	min	10.13	9.11	7.85
				max	20.32	17.63	17.07
# de solutions avec une RMSD < 10Å (sur 50 solutions)					0	1	6
4WFL	107	18	1.8	min	14.70	9.79	8.82
				max	21.38	17.21	16.22
# de solutions avec une RMSD < 10Å (sur 50 solutions)					0	1	5
4QK8	124	20	1.3	min	13.23	12.25	13.70
				max	22.19	22.78	23.13
# de solutions avec une RMSD < 14Å (sur 50 solutions)					1	3	1
1MFQ	127	24	1.44	min	14.89	10.49	9.68
				max	25.59	21.05	20.64
# de solutions avec une RMSD < 11Å (sur 50 solutions)					0	3	7

TABLE AT.4 – **Comparaison des RMSD entre les différentes familles de paramètres.** Le tableau compare les structures calculées pour chaque molécule d'après les différentes familles de paramètres possibles pour l'*ensemble de test*. Les résultats des RMSD sont calculés sur 50 structures. Nous indiquons en bleu la famille choisie automatiquement (via notre méthode) pour la molécule. La famille que nous choisissons est celle permettant d'avoir une RMSD minimale et plusieurs structures avec des RMSD faibles. Notre choix de famille (entre les trois proposées) est donc pertinent.

ID	GARN		MC-Sym		Farna		NAST		iFoldRNA	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
1E8O	3.75	9.07	1.65	11.52	4.12	11.96	5.57	9.82	6.45	10.60
4FE5	2.79	4.77	–	–	1.23	5.87	1.95	12.35	4.80	11.12
4QJH	4.15	7.02	–	–	2.41	12.40	9.24	12.38	7.35	17.16
1LNG	7.19	7.19	3.76	6.85	3.02	10.31	6.05	36.56	3.79	9.67
4WFL 1	6.79	6.79	–	–	3.24	8.34	14.01	17.24	8.98	17.57
4WFL 2	5.65	8.83	–	–	4.64	12.25	11.26	14.40	7.46	14.11
4QK8 1	5.12	7.87	–	–	2.94	8.64	5.16	10.68	6.21	13.25
4QK8 2	3.95	8.29	–	–	3.82	10.13	8.62	12.20	5.35	14.75
1MFQ	4.70	5.71	2.87	6.06	5.31	7.5	9.26	19.71	18.16	27.13

TABLE AT.5 – **Comparaison avec les approches de l'état de l'art pour le repliement des 3-jonctions de notre *ensemble de test*.** Le tableau présente les valeurs des RMSD sur des 3-jonctions de notre *ensemble de test* pour notre méthode préliminaire de GARN et pour les autres approches à grain fin de l'état de l'art. Les RMSD sont calculées sur les joueurs de la 3-jonction et sur les joueurs adjacents à la 3-jonction. S'il y a plusieurs 3-jonctions, nous indiquons la jonction d'après l'ordre des joueurs. Farna permet d'avoir des RMSD au niveau de la 3-jonction très faible. Pour la majorité des jonctions, la RMSD minimale atteinte par GARN est d'au maximum 4 Å de plus que la RMSD minimale pour toutes les approches.

ID	# Nucl.	# Joueurs	GARN min	RNAJAG min
2FK6	52	11	9.56	4.01
1DK1	57	13	4.58	6.16
1MMS	58	11	4.21	4.13
3EGZ	65	10	6.57	6.59
3D2G	77	14	4.67	2.07
2HOJ	78	15	1.99	2.18
2GDI	80	15	4.58	1.98

TABLE AT.6 – **Comparaison des RMSD des 3-jonctions avec RNAJAG.** Le tableau présente les valeurs des RMSD sur des 3-jonctions pour notre méthode préliminaire de GARN et pour RNAJAG. Les valeurs de RNAJAG sont prises de [Kim *et al.*, 2014]. Les RMSD de GARN sont calculées sur les joueurs de la 3-jonction et sur les joueurs adjacents à la 3-jonction. Les RMSD en bleu indiquent les valeurs les plus faibles. Alors que RNAJAG se concentre sur le repliement des jonctions, nous n'imposons aux 3-jonctions que l'empilement entre deux hélices. Nos résultats sur les 3-jonctions sont comparables à ceux de RNAJAG, nos deux méthodes fournissant des 3-jonctions avec des RMSD faibles pour certaines molécules. Par exemple, GARN possède de meilleurs 3-jonctions pour la molécule 1DK1, alors que RNAJAG propose de meilleurs 3-jonctions pour la molécule 2FK6.

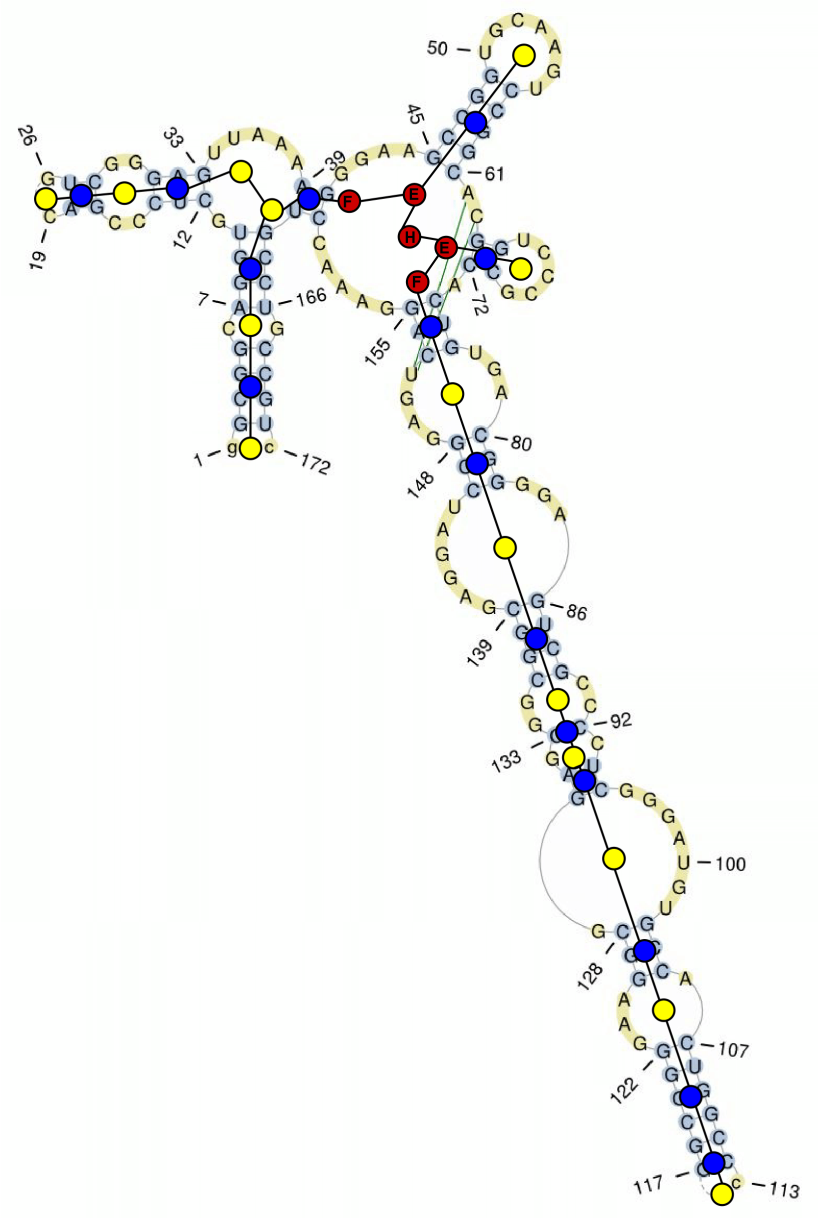


FIGURE AF.2 – **Modèle de représentation de 4GXY avec deux 3-jonctions.** La molécule 4GXY, possédant une 4-jonction, a été modélisation avec deux 3-jonctions pour notre méthode préliminaire. Nous représentons en rouge les joueurs de cette 4-jonction. Les joueurs possédant trois arêtes sont les joueurs d'empilements (noté E) des 3-jonctions. Nous ajoutons un joueur hélice (H). Les joueurs représentant la famille des 3-jonctions sont notés (F).

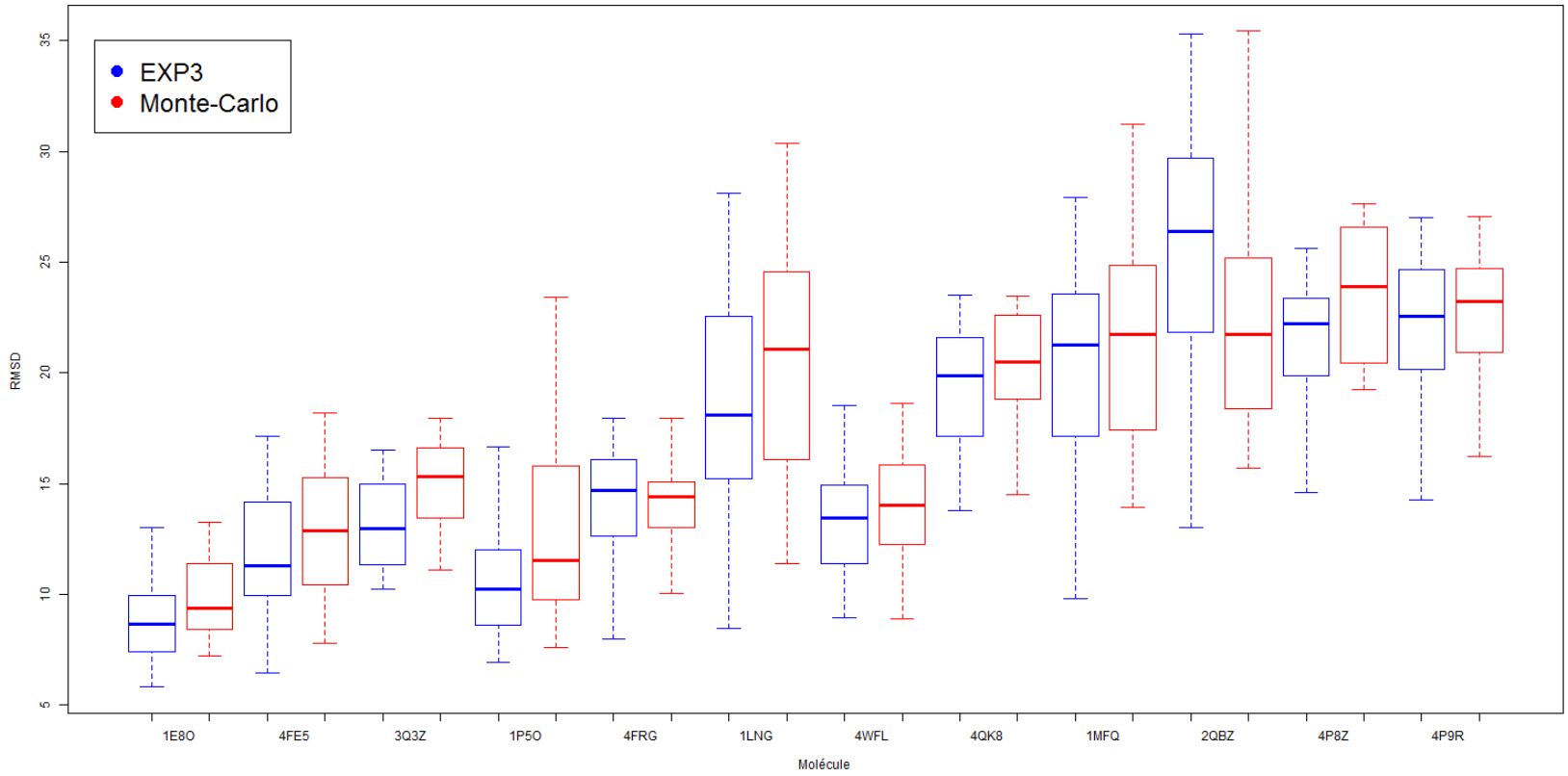


FIGURE AF.3 – **Répartition des RMSD avec l'algorithme EXP3 ou avec la méthode de Monte-Carlo.** La méthode de Monte-Carlo a été testé sur GARN, sur notre *ensemble de test élargi*. Les résultats du graphe pour plusieurs molécules de cet ensemble montre que la méthode de Monte-Carlo donne, en général, des ensembles avec une moyenne de RMSD plus élevée (donc des structures moins correctes) que l'algorithme EXP3. Dans tous les cas, la structure de RMSD minimale est trouvée avec EXP3.

ID	Description	# Nucl.	# Joueurs	# jonctions ≥ 3
1E8O	7SL RNA	50	8	1
1MZP	Fragment of 23S rRNA	55	8	0
4FE5	Guanine riboswitch aptamer domain	68	14	1
2DU3	O-phosphoseryl-tRNA synthetase	71	13	1
4QJH	Twister Ribozyme	74	15	1
3Q3Z	c-di-GMP-II riboswitch	77	9	0
1P5O	77-MER	77	19	0
2HOJ	Thi-box riboswitch	79	15	1
4FRG	Cobalamin riboswitch aptamer domain	84	17	1
4TS0	Spinach RNA aptamer	89	21	1
2GIS	SAM-I riboswitch	94	21	1
1LNG	7S.S SRP RNA	97	16	1
4WFL	Bacterial SRP Alu domain	107	18	2
1C2X	5S RIBOSOMAL RNA	120	17	1
4QK8	C-di-AMP riboswitch	124	20	2
1MFQ	7S RNA of human SRP	127	24	1
1GID	P4-P6 RNA Ribozyme Domain	158	27	1
3D0U	Lysine Riboswitch RNA	161	32	1
2QBZ	M-Box RNA, ykoK riboswitch aptamer	161	28	2
4GXY	Adenosylcobalamin riboswitch	172	34	2
4P8Z	18S rRNA gene	188	29	4
4P9R	RNA (189-MER)	192	30	4
4GMA	Adenosylcobalamin riboswitch	210	36	3
4C4Q	Internal Ribosomal Entry Site	233	49	3
3DHS	RNase P RNA	268	42	2

TABLE AT.7 – **Ensemble de test élargi.** L'*ensemble de test élargi* contient 25 molécules de tailles différentes, dont les molécules de l'*ensemble de test* du chapitre 4 (voir la table en annexe AT.2). Nous précisons ici la présence 3-jonctions ou de jonctions plus larges. Par exemple, la molécule 3DHS possède une 3-jonction et une 6-jonction.

ID	# Nucl.	GARN		FARNA		NAST		iFoldRNA		RNAComposer		SiMRNA		RNAJag	ERNWIN
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
1E8O	50	5.8	13.02	8.72	18.62	17.44	23.52	10.09	21.69	0.50	1.82	2.95	4.39	–	7.63
1MZP	55	4.2	11.03	5.28	14.37	13.76	19.29	7.73	19.09	0.91	4.20	6.47	9.02	6.74	4.09
4FE5	68	6.43	18.85	7.63	19.56	17.37	26.20	13.03	19.24	1.42	4.71	4.52	6.16	–	4.75
2DU3	71	10.14	20.54	7.06	15.59	15.85	131.06	9.97	15.69	1.49	3.31	5.83	10.73	–	9.80
4QJH	74	6.44	18.9	9.19	18.69	21.68	25.55	11.23	20.58	9.05	12.14	6.34	15.17	–	–
3Q3Z	77	10.24	18.91	10.54	21.55	27.95	33.92	9.22	18.66	9.05	9.94	15.17	18.16	–	7.67
1P5O	77	6.9	20.72	7.22	22.70	15.44	20.31	15.26	21.50	3.67	7.86	7.59	10.47	–	9.36
2HOJ	79	6.62	19.11	6.94	21.69	25.41	29.96	10.26	19.15	16.37	19.65	8.28	14.33	13.01	6.64
4FRG	84	7.99	19.83	8.95	20.15	28.56	31.93	11.61	22.96	1.40	3.61	5.67	13.85	–	11.95
4TS0	89	7.47	15.47	8.04	19.30	28.56	31.93	–	–	6.59	12.79	–	–	–	–
2GIS	94	10.72	17.13	10.42	24.30	37.84	40.77	13.06	28.78	3.87	9.90	13.68	20.00	–	11.80
1LNG	97	8.45	28.08	15.16	32.00	43.84	49.11	11.17	34.29	2.51	8.82	18.29	24.14	14.56	7.08
4WFL	107	8.93	19.52	10.76	24.13	41.74	49.03	15.12	25.56	6.45	9.90	16.64	20.61	–	13.63
1C2X	120	11.45	25.12	–	–	–	–	14.14	19.95	4.47	8.72	12.53	21.85	–	27.35
4QK8	124	13.79	24.5	14.07	27.05	53.47	59.50	18.40	27.51	10.66	19.41	19.36	23.20	–	13.87
1MFQ	127	9.62	30.32	–	–	56.94	60.83	21.01	30.15	3.69	7.04	9.50	30.45	10.55	11.91
1GID	158	16.36	37.72	17.47	47.02	80.57	83.57	26.47	42.73	6.62	17.43	27.45	41.69	28.24	25.2
3D0U	161	12.4	29.08	–	–	77.81	83.01	22.86	47.16	3.47	14.57	15.82	21.92	–	21.04
2QBZ	161	13	36.69	15.29	39.47	83.44	88.87	21.89	35.67	4.09	13.19	22.65	29.33	–	18.85
4GXY	172	14.75	29.41	19.60	34.41	89.06	94.04	–	–	4.86	16.63	27.82	36.05	–	16.0
4P8Z	188	14.6	27.55	18.60	33.84	97.73	103.73	–	–	26.78	30.59	22.63	36.08	–	21.63
4P9R	192	14.23	30.55	17.04	40.40	100.30	109.79	–	–	23.76	31.88	13.32	29.47	–	–
4GMA	210	21.33	34.17	21.63	43.89	116.65	121.12	116.65	121.12	17.20	25.32	–	–	–	24.8
4C4Q	233	16.32	39.43	19.71	47.70	112.48	118.27	–	–	6.92	50.01	–	–	–	28.57
3DHS	268	16.54	28.75	19.67	38.46	162.63	168.09	–	–	7.16	8.37	–	–	–	19.9

TABLE AT.8 – **Résultats des structures des approches de l'état de l'art pour l'*ensemble de test élargi*.** Le tableau compare GARN avec les structures fournies par iFoldRNA, FARNA, NAST, RNAComposer, RNAJAG, ERNWIN et SimRNA. Nous notons en bleu les RMSD minimales atteintes (en dehors de RNAComposer, qui, grâce à sa méthode, obtiendra toujours de meilleurs résultats dans le cas de nos molécules). Certaines méthodes ne peuvent pas générer de solutions pour tout l'*ensemble de test élargi*. Les résultats de RNAJAG proviennent de la littérature [Kim *et al.*, 2014]. Ceux de ERNWIN proviennent de leur logiciel qui fournit une structure d'énergie minimale. Les résultats de SimRNA proviennent de leur serveur [Magnus *et al.*, 2016] qui ne renvoie que 3 structures. Nous ne notons donc ici Notre méthode donne de bons résultats globaux pour les structures, obtenant la structure de RMSD minimale pour toutes les molécules possédant plus de 100 nucléotides.

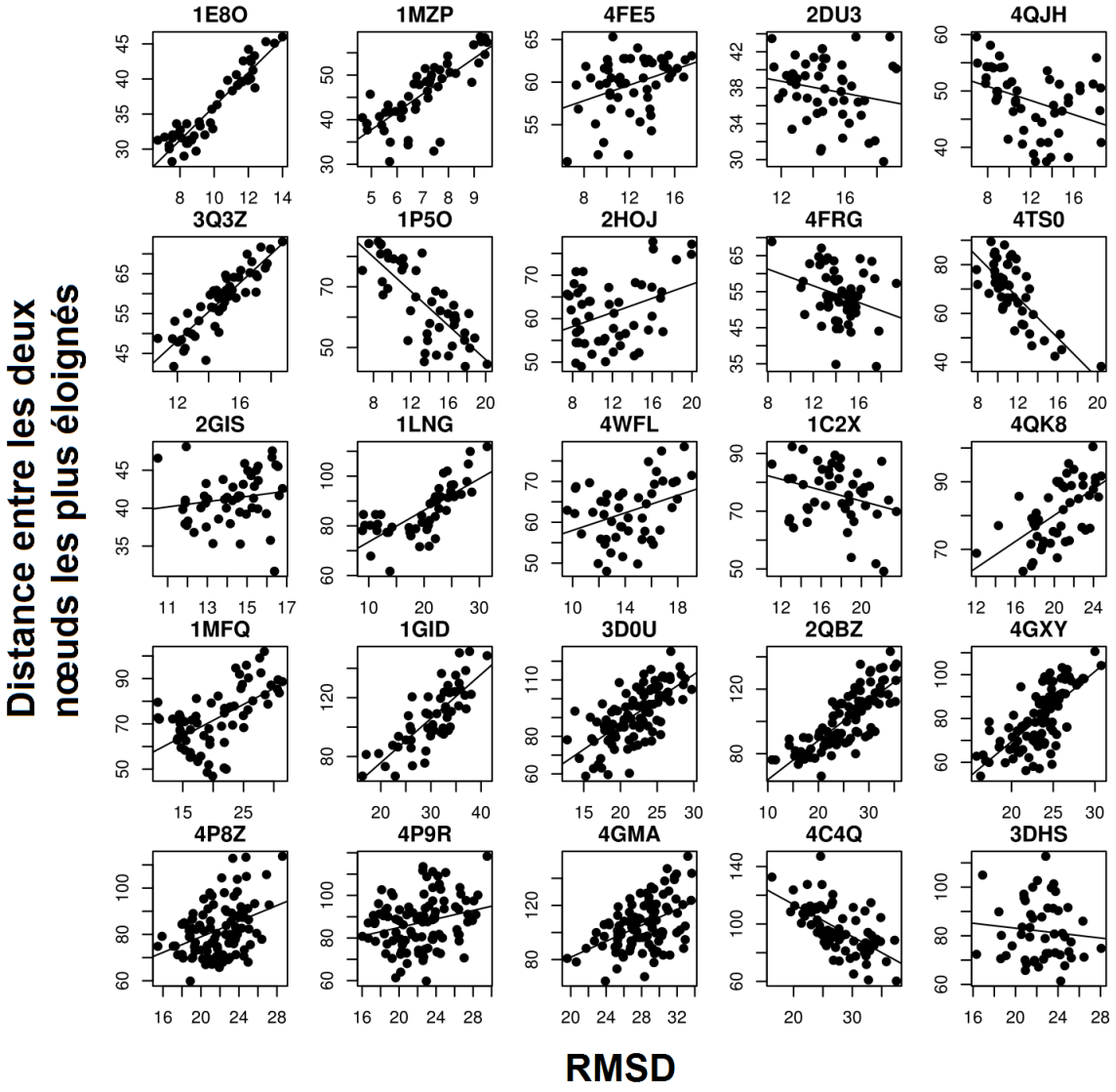


FIGURE AF.4 – **Rapport entre les RMSD et les distances entre les deux nœuds les plus éloignés des structures pour notre *ensemble de test élargi*.** Les figures représentent le rapport entre les RMSD trouvées (en abscisse) et la distance entre les deux nœuds les plus éloignés de la structure (en ordonnée), pour chaque molécule de notre *ensemble de test élargi*, avec un échantillonnage de 50 structures. La ligne représente la régression linéaire de nos variables. Nous observons que cette régression linéaire est diagonale pour la majorité des molécules.

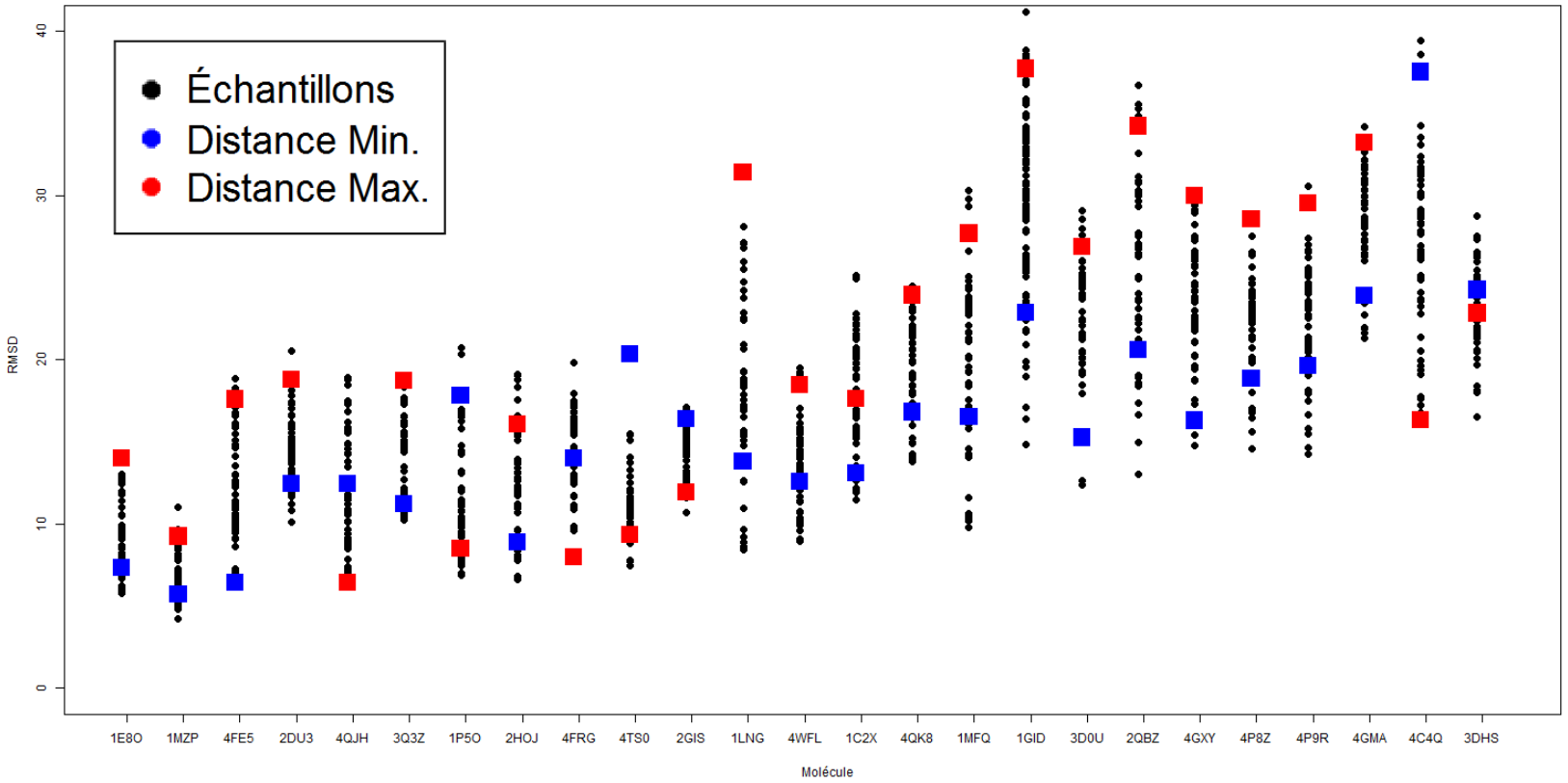


FIGURE AF.5 – **Visualisation de l'espace de solutions et du tri pour GARN sur notre *ensemble de test élargi*.** Sur l'échantillon de structures proposé par GARN, le tri renvoie deux structures possibles : l'une avec une distance minimale entre les deux nœuds les plus éloignés (en bleu) et une avec cette distance maximisée (en rouge). L'une de ces deux structures se trouvent, pour la majorité des molécules, dans le premier quartile, ou représente la meilleure structure.

ID PDB	FARNA			NAST			iFoldRNA			RNAComposer		
	1^{er} Q.	Min	Max	1^{er} Q.	Min	Max	1^{er} Q.	Min	Max	1^{er} Q.	Min	Max
1E8O	11.23	9.74	17.72	19.34	17.44	23.49	12.95	11.56	21.69	0.78	0.74	1.82
1MZP	7.92	7.60	14.37	16.05	15.55	18.45	13.62	7.73	19.09	2.16	1.88	4.2
4FE5	12.33	12.21	19.39	22.92	17.37	26.2	15.29	13.97	17.11	1.84	2.71	2.4
2DU3	11.36	12.73	7.96	18.89	17.79	131.06	11.48	15.69	12.67	1.69	2.47	2.67
4QJH	11.95	15.41	15.29	23.24	23.07	24.77	15.77	13.96	20.58	10.15	11.45	9.47
3Q3Z	13.38	12.97	20.95	30.78	27.95	33.92	11.13	10.26	18.66	9.33	9.37	9.94
1P5O	11.67	22.7	11.50	17.11	16.38	18.93	18.97	20.18	15.26	4.99	6.82	7.86
2HOJ	12.72	10.45	21.69	26.88	26.15	29.96	15.04	13.70	18.11	17.13	16.49	19.65
4FRG	13.16	15.86	9.54	29.72	29.29	31.42	16.22	17.06	22.87	2.57	2.6	3.61
4TS0	10.44	16.48	12.52	–	–	–	–	–	–	8.16	8.72	9.79
2GIS	15.9	11.78	20.22	38.8	37.84	39.97	17.67	15.36	28.78	4.55	9.9	4.14
1LNG	20.27	15.78	32	45.47	44.58	47.21	16.99	11.96	30.44	6.69	6.59	7.24
4WFL	15.98	15.04	20.53	44.13	43.28	47.02	19.99	21.47	15.12	6.69	7.23	6.49
1C2X	–	–	–	–	–	–	15.57	19.67	18.76	6.08	8.72	6.59
4QK8	18.49	14.08	22.93	54.8	54.29	57.2	19.62	18.95	27.51	12.66	12.66	14.06
1MFQ	20.18	29.57	27.76	57.79	57.65	60.83	26.79	27.49	24.37	5.24	6.13	7.04
1GID	22.01	21.70	45.73	83.40	80.57	86.51	32.68	29.86	42.73	6.79	11.33	17.43
3D0U	–	–	–	79.46	78.36	82.33	38.33	29.71	45.48	7.46	3.47	13.17
2QBZ	23.63	24.06	30.09	86.38	84.13	87.68	26.72	26.08	31.49	5.45	8.45	4.81
4GXY	23.33	19.60	32.85	91.83	89.06	93.04	–	–	–	8.36	4.86	12.29
4P8Z	22.77	20.34.	25.54	100.11	101.05	100.99	–	–	–	29.07	26.78	29.07
4P9R	25.51	20.49	40.4	106.49	100.30	108.58	–	–	–	26.47	25.49	26.58
4GMA	25.53	20.53	43.89	117.75	117.19	120	–	–	–	20.3	17.20	25.32
4C4Q	24.75	30.74	47.7	114.41	112.94	118.27	–	–	–	9.57	9.16	50.01
3DHS	25	24.66	38.46	163.63	163.31	168.09	–	–	–	7.56	8.16	7.16

TABLE AT.9 – **Résultats du tri par distance pour les approches existantes sur notre *ensemble de test élargi*.** Ce tableau compare la RMSD pour la structure ayant une distance minimale (Min) ou maximale (Max) entre les nœuds les plus éloignés, et le 1^{er} quartile (1^{er} Q.) pour les RMSD des structures de l'*ensemble de test élargi*. En bleu les structures avec une RMSD inférieure au 1^{er} quartile. Nous pouvons trier NAST en utilisant la distance minimale. Pour FARNA et iFoldRNA, l'une des deux structures proposées a une RMSD faible.

Table des figures

Liste des tableaux

Bibliographie

[Auer *et al.*, 2002a] Auer, P., Cesa-Bianchi, N. et Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

[Auer *et al.*, 2002b] Auer, P., Cesa-Bianchi, N., Freund, Y. et Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

[Aumann, 1974] Aumann, R. J. (1974). Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96.

[Batey *et al.*, 1999] Batey, R. T., Rambo, R. P. et Doudna, J. A. (1999). Tertiary motifs in RNA structure and folding. *Angewandte Chemie International Edition*, 38(16):2326–2343.

[Berenbrink *et al.*, 2007] Berenbrink, P., Friedetzky, T., Hajirasouliha, I. et Hu, Z. (2007). Convergence to equilibria in distributed, selfish reallocation processes with weighted tasks. *In Algorithms-European Symposium on Algorithms 2007*, pages 41–52. Springer.

[Berenbrink et Schulte, 2007] Berenbrink, P. et Schulte, O. (2007). Evolutionary equilibrium in bayesian routing games : Specialization and niche formation. *In ESA*, pages 29–40.

[Bernauer *et al.*, 2011] Bernauer, J., Huang, X., Sim, A. Y. L. et Levitt, M. (2011). Fully differentiable coarse-grained and all-atom knowledge-based potentials for RNA structure evaluation. *RNA*, 17(6):1066–1075.

[Boniecki *et al.*, 2015] Boniecki, M. J., Lach, G., Dawson, W. K., Tomala, K., Lukasz, P., Soltysinski, T., Rother, K. M. et Bujnicki, J. M. (2015). SimRNA : a coarse-grained method for RNA folding simulations and 3D structure prediction. *Nucleic Acids Research*, 44(7):e63–e63.

[Boudard *et al.*, 2015] Boudard, M., Bernauer, J., Barth, D., Cohen, J. et Denise, A. (2015). GARN : Sampling RNA 3D structure space with game theory and knowledge-based scoring strategies. *PloS ONE*, 10(8):e0136444.

[Brion et Westhof, 1997] Brion, P. et Westhof, E. (1997). Hierarchy and dynamics of RNA folding. *Annual Review of Biophysics and Biomolecular Structure*, 26:113–137.

[Cesa-Bianchi et Lugosi, 2006] Cesa-Bianchi, N. et Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

[Chastain et Tinoco Jr, 1991] Chastain, M. et Tinoco Jr, I. (1991). Structural elements in rna. *Progress in Nucleic Acid Research and Molecular Biology*, 41:131–177.

[Cooper *et al.*, 2009] Cooper, T. A., Wan, L. et Dreyfuss, G. (2009). RNA and disease. *Cell*, 136(4):777–793.

[Crick, 1966] Crick, F. (1966). Codon-anticodon pairing : the wobble hypothesis. *Journal of Molecular Biology*, 19(2):548 – 555.

[Cruz *et al.*, 2012] Cruz, J. A., Blanchet, M.-F., Boniecki, M., Bujnicki, J. M., Chen, S.-J., Cao, S., Das, R., Ding, F., Dokholyan, N. V., Flores, S. C., Huang, L., Lavender, C. A., Lisi, V., Major, F., Mikolajczak, K., Patel, D. J., Philips, A., Puton, T., Santalucia, J., Sijenyi, F., Hermann, T., Rother, K., Rother, M., Serganov, A., Skorupski, M., Soltysinski, T., Sripakdeevong, P., Tuszynska, I., Weeks, K. M., Waldsich, C., Wildauer, M., Leontis, N. B. et Westhof, E. (2012). RNA-Puzzles : a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*, 18(4):610–625.

[Das et Baker, 2007] Das, R. et Baker, D. (2007). Automated de novo prediction of native-like RNA tertiary structures. *Proceedings of the National Academy of Sciences*, 104(37):14664–14669.

[Dima *et al.*, 2005] Dima, R. I., Hyeon, C. et Thirumalai, D. (2005). Extracting stacking interaction parameters for RNA from the data set of native structures. *Journal of Molecular Biology*, 347(1):53–69.

[Ding *et al.*, 2008] Ding, F., Sharma, S., Chalasani, P., Demidov, V. V., Broude, N. E. et Dokholyan, N. V. (2008). Ab initio RNA folding by discrete molecular dynamics : from structure prediction to folding mechanisms. *RNA*, 14(6):1164–1173.

[Djelloul et Denise, 2008] Djelloul, M. et Denise, A. (2008). Automated motif extraction and classification in RNA tertiary structures. *RNA*, 14(12):2489–2497.

[Doty *et al.*, 1959] Doty, P., Boedtker, H., Fresco, J. et Haselkorn, R and, L. M. (1959). Secondary structure in ribonucleic acids. *Proceedings of the National Academy of Sciences of the United States of America*, 45(4):482.

[Fire *et al.*, 1998] Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. E. et Mello, C. C. (1998). Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. *nature*, 391(6669):806–811.

[Flores et Altman, 2010] Flores, S. C. et Altman, R. B. (2010). Turning limited experimental information into 3D models of RNA. *RNA*, 16(9):1769–1778.

[Fonseca *et al.*, 2014] Fonseca, R., Pachov, D. V., Bernauer, J. et van den Bedem, H. (2014). Characterizing RNA ensembles from NMR data with kinematic models. *Nucleic Acids Research*, 42(15):9562–9572.

[Frellsen *et al.*, 2009] Frellsen, J., Moltke, I., Thiim, M., Mardia, K. V., Ferkinghoff-Borg, J. et Hamelryck, T. (2009). A probabilistic model of RNA conformational space. *PLOS Computational Biology*, 5(6):e1000406.

[Freund *et al.*, 1997] Freund, Y., Schapire, R. E., Singer, Y. et Warmuth, M. K. (1997). Using and combining predictors that specialize. *In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 334–343. ACM.

[Gillespie *et al.*, 2009] Gillespie, J., Mayne, M. et Jiang, M. (2009). RNA folding on the 3D triangular lattice. *BMC Bioinformatics*, 10:369.

[Grigoriadis et Khachiyan, 1995] Grigoriadis, M. D. et Khachiyan, L. G. (1995). A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters*, 18(2):53–58.

[Gros *et al.*, 1961] Gros, F., Hiatt, H., Gilbert, W., Kurland, C. G., Risebrough, R. et Watson, J. D. (1961). Unstable ribonucleic acid revealed by pulse labelling of escherichia coli.

[Guo, 2010] Guo, P. (2010). The emerging field of RNA nanotechnology. *Nature Nanotechnology*, 5(12):833–842.

[Hart et Mascolell, 2000] Hart, S. et Mascolell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):11271150.

[Hofacker, 2009] Hofacker, I. L. (2009). RNA secondary structure analysis using the vienna RNA package. *Current Protocols in Bioinformatics*, Chapter 12:Unit12.2.

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

[Istrail *et al.*, 2009] Istrail, S., Lam, F. *et al.* (2009). Combinatorial algorithms for protein folding in lattice models : a survey of mathematical results. *Communications in Information & Systems*, 9(4):303–346.

[Jacob et Monod, 1961] Jacob, F. et Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3(3):318–356.

[Jean-Yves Audibert and, 2009] Jean-Yves Audibert and, S. B. (2009). Minimax policies for adversarial and stochastic bandits. *In COLT 2009*.

[Jonikas *et al.*, 2009] Jonikas, M. A., Radmer, R. J., Laederach, A., Das, R., Pearlman, S., Herschlag, D. et Altman, R. B. (2009). Coarse-grained modeling of large RNA molecules with knowledge-based potentials and structural filters. *RNA*, 15(2):189–199.

[Jossinet *et al.*, 2010] Jossinet, F., Ludwig, T. E. et Westhof, E. (2010). Assemble : an interactive graphical tool to analyze and build RNA architectures at the 2D and 3D levels. *Bioinformatics*, 26(16):2057–2059.

[Kabsch, 1976] Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A : Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923.

[Kerpedjiev *et al.*, 2015] Kerpedjiev, P., Zu Siederdissen, C. H. et Hofacker, I. L. (2015). Predicting RNA 3D structure using a coarse-grain helix-centered model. *RNA*, 21(6):1110–1121.

[Kim *et al.*, 2014] Kim, N., Laing, C., Elmetwaly, S., Jung, S., Curuksu, J. et Schlick, T. (2014). Graph-based sampling for approximating global helical topologies of RNA. *Proceedings of the National Academy of Sciences*, 111(11):4079–4084.

[Koutsoupias et Papadimitriou, 1999] Koutsoupias, E. et Papadimitriou, C. (1999). Worst-case equilibria. *In STACS 1999*, pages 404–413. Springer.

[Krichene *et al.*, 2015] Krichene, W., Drighès, B. et Bayen, A. M. (2015). Online learning of nash equilibria in congestion games. *SIAM Journal on Control and Optimization*, 53(2):1056–1081.

[Laing *et al.*, 2009] Laing, C., Jung, S., Iqbal, A. et Schlick, T. (2009). Tertiary motifs revealed in analyses of higher-order RNA junctions. *Journal of Molecular Biology*, 393(1):67–82.

[Laing *et al.*, 2013] Laing, C., Jung, S., Kim, N., Elmetwaly, S., Zahran, M. et Schlick, T. (2013). Predicting helical topologies in RNA junctions as tree graphs. *PLoS One*, 8(8):e71947.

[Laing et Schlick, 2009] Laing, C. et Schlick, T. (2009). Analysis of four-way junctions in rna structures. *Journal of molecular biology*, 390(3):547–559.

[Laing et Schlick, 2010] Laing, C. et Schlick, T. (2010). Computational approaches to 3D modeling of RNA. *Journal of Physics : Condensed Matter*, 22(28):283101.

[Laing et Schlick, 2011] Laing, C. et Schlick, T. (2011). Computational approaches to RNA structure prediction, analysis, and design. *Current Opinion in Structural Biology*, 21(3):306–318.

[Laing *et al.*, 2012] Laing, C., Wen, D., Wang, J. T. et Schlick, T. (2012). Predicting coaxial helical stacking in rna junctions. *Nucleic Acids Research*, 40(2):487–498.

[Lamiable *et al.*, 2012] Lamiable, A., Barth, D., Denise, A., Quessette, F., Vial, S. et Westhof, E. (2012). Automated prediction of three-way junction topological families in RNA secondary structures. *Computational Biology and Chemistry*, 37:1–5.

[Lamiable *et al.*, 2013] Lamiable, A., Quessette, F., Vial, S., Barth, D. et Denise, A. (2013). An algorithmic game-theory approach for coarse-grain prediction of RNA 3D structure. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(1):193–199.

[Le *et al.*, 1989] Le, S. Y., Nussinov, R. et Maizel, J. V. (1989). Tree graphs of RNA secondary structures and their comparisons. *Computers and Biomedical Research*, 22(5):461–473.

[Leontis *et al.*, 2006] Leontis, N. B., Lescoute, A. et Westhof, E. (2006). The building blocks and motifs of RNA architecture. *Current Opinion in Structural Biology*, 16(3):279–287.

[Leontis et Westhof, 2001] Leontis, N. B. et Westhof, E. (2001). Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512.

[Lescoute *et al.*, 2005] Lescoute, A., Leontis, N. B., Massire, C. et Westhof, E. (2005). Recurrent structural RNA motifs, isostericity matrices and sequence alignments. *Nucleic Acids Research*, 33(8):2395–2409.

[Lescoute et Westhof, 2006] Lescoute, A. et Westhof, E. (2006). Topology of three-way junctions in folded RNAs. *RNA*, 12(1):83–93.

[Li, 2013] Li, P. T. (2013). Analysis of diffuse k+ and mg2+ ion binding to a two-base-pair kissing complex by single-molecule mechanical unfolding. *Biochemistry*, 52(29):4991–5001.

[Magnus *et al.*, 2016] Magnus, M., Boniecki, M., Dawson, W. et Bujnicki, J. (2016). SimRNAweb : a web server for RNA 3D structure modeling with optional restraints. *Nucleic acids research*.

[Maiorov et Crippen, 1994] Maiorov, V. N. et Crippen, G. M. (1994). Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *Journal of molecular biology*, 235(2):625–634.

[Mandal et Breaker, 2004] Mandal, M. et Breaker, R. R. (2004). Gene regulation by riboswitches. *Nature Reviews Molecular Cell Biology*, 5(6):451–463.

[Martinez *et al.*, 2008] Martinez, H. M., Maizel, Jr, J. V. et Shapiro, B. A. (2008). RNA2D3D : a program for generating, viewing, and comparing 3-dimensional models of RNA. *Journal of Biomolecular Structure and Dynamics*, 25(6):669–683.

[Mathews, 2006] Mathews, D. H. (2006). Revolutions in RNA secondary structure prediction. *Journal of Molecular Biology*, 359(3):526–532.

[Meister et Tuschl, 2004] Meister, G. et Tuschl, T. (2004). Mechanisms of gene silencing by double-stranded RNA. *Nature*, 431(7006):343–349.

[Metropolis, 1987] Metropolis, N. (1987). The beginning of the monte carlo method. *Los Alamos Science*, 15(584):125–130.

[Metropolis et Ulam, 1949] Metropolis, N. et Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247):335–341.

[Murray *et al.*, 2003] Murray, L. J. W., Arendall, 3rd, W. B., Richardson, D. C. et Richardson, J. S. (2003). RNA backbone is rotameric. *Proceedings of the National Academy of Sciences*, 100(24):13904–13909.

[Nasalean *et al.*, 2009] Nasalean, L., Stombaugh, J., Zirbel, C. L. et Leontis, N. B. (2009). RNA 3D structural motifs : definition, identification, annotation, and database searching. *In Non-protein coding RNAs*, pages 1–26. Springer.

[Nash, 1950] Nash, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49.

[Nussinov *et al.*, 1978] Nussinov, R., Pieczenik, G., Griggs, J. R. et Kleitman, D. J. (1978). Algorithms for loop matchings. *SIAM Journal on Applied mathematics*, 35(1):68–82.

[Papadimitriou, 1994] Papadimitriou, C. H. (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532.

[Papadimitriou, 2007] Papadimitriou, C. H. (2007). *The complexity of finding Nash equilibria*. Cambridge University Press.

[Parisien et Major, 2008] Parisien, M. et Major, F. (2008). The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51–55.

[Parker et Song, 2004] Parker, R. et Song, H. (2004). The enzymes and control of eukaryotic mrna turnover. *Nature structural & molecular biology*, 11(2):121–127.

[Popenda *et al.*, 2008] Popenda, M., Błażewicz, M., Szachniuk, M. et Adamiak, R. W. (2008). RNA FRABASE version 1.0 : an engine with a database to search for the three-dimensional fragments within RNA structures. *Nucleic Acids Research*, 36(suppl 1):D386–D391.

[Popenda *et al.*, 2012] Popenda, M., Szachniuk, M., Antczak, M., Purzycka, K. J., Lukasiak, P., Bartol, N., Blazewicz, J. et Adamiak, R. W. (2012). Automated 3d structure composition for large RNAs. *Nucleic Acids Research*, page gks339.

[Popenda *et al.*, 2010] Popenda, M., Szachniuk, M., Blazewicz, M., Wasik, S., Burke, E. K., Blazewicz, J. et Adamiak, R. W. (2010). RNA FRABASE 2.0 : an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures. *BMC Bioinformatics*, 11:231.

[Reeder *et al.*, 2006] Reeder, J., Höchsmann, M., Rehmsmeier, M., Voss, B. et Giegerich, R. (2006). Beyond Mfold : recent advances in RNA bioinformatics. *Journal of Biotechnology*, 124(1):41–55.

[Robbins, 1985] Robbins, H. (1985). Some aspects of the sequential design of experiments. *In* Lai, T. et Siegmund, D., éditeurs : *Herbert Robbins Selected Papers*, pages 169–177. Springer New York.

[Rosenthal, 1973] Rosenthal, R. (1973). A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67.

[Rother *et al.*, 2011] Rother, M., Rother, K., Puton, T. et Bujnicki, J. M. (2011). RNA tertiary structure prediction with ModeRNA. *Briefings in Bioinformatics*, 12(6):601–613.

[Roughgarden, 2005] Roughgarden, T. (2005). *Selfish routing and the price of anarchy*. The MIT Press.

[Sastry *et al.*, 1994] Sastry, P., Phansalkar, V. et Thathachar, M. (1994). Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(5):769–777.

[Schudoma *et al.*, 2011] Schudoma, C., Larhlimi, A. et Walther, D. (2011). The influence of the local sequence environment on RNA loop structures. *RNA*, 17(7):1247–1257.

[Shapiro *et al.*, 2007] Shapiro, B. A., Yingling, Y. G., Kasprzak, W. et Bindewald, E. (2007). Bridging the gap in RNA structure prediction. *Current Opinion in Structural Biology*, 17(2):157–165.

[Shapiro et Zhang, 1990] Shapiro, B. A. et Zhang, K. Z. (1990). Comparing multiple RNA secondary structures using tree comparisons. *Computer Applications in the Biosciences*, 6(4):309–318.

[Sharma *et al.*, 2008] Sharma, S., Ding, F. et Dokholyan, N. V. (2008). iFoldRNA : three-dimensional RNA structure prediction and folding. *Bioinformatics*, 24(17):1951–1952.

[Sheikh *et al.*, 2012] Sheikh, S., Backofen, R. et Ponty, Y. (2012). Impact of the energy model on the complexity of rna folding with pseudoknots. *In Combinatorial Pattern Matching*, pages 321–333. Springer.

[Sim *et al.*, 2012a] Sim, A. Y. L., Levitt, M. et Minary, P. (2012a). Modeling and design by hierarchical natural moves. *Proceedings of the National Academy of Sciences*, 109(8):2890–2895.

[Sim *et al.*, 2012b] Sim, A. Y. L., Minary, P. et Levitt, M. (2012b). Modeling nucleic acids. *Current Opinion in Structural Biology*, 22(3):273–278.

[Sinden, 2012] Sinden, R. R. (2012). *DNA structure and function*. Elsevier.

[Smith, 1979] Smith, J. M. (1979). Game theory and the evolution of behaviour. *Proceedings of the Royal Society of London B : Biological Sciences*, 205(1161):475–488.

[Smith, 1982] Smith, J. M. (1982). *Evolution and the Theory of Games*. Cambridge University Press.

[Staple et Butcher, 2005] Staple, D. W. et Butcher, S. E. (2005). Pseudoknots : RNA structures with diverse functions. *PLOS Biology*, 3(6):e213.

[Sutton et Barto, 1998] Sutton, R. S. et Barto, A. G. (1998). *Introduction to reinforcement learning*, volume 135. MIT Press Cambridge.

[Sykes et Levitt, 2005] Sykes, M. T. et Levitt, M. (2005). Describing RNA structure by libraries of clustered nucleotide doublets. *Journal of Molecular Biology*, 351(1):26–38.

[Thirumalai et Hyeon, 2009] Thirumalai, D. et Hyeon, C. (2009). *Theory of RNA folding : from hairpins to ribozymes*. Springer.

[Tinoco et Bustamante, 1999] Tinoco, Jr, I. et Bustamante, C. (1999). How RNA folds. *Journal of Molecular Biology*, 293(2):271–281.

[Tyagi et Mathews, 2007] Tyagi, R. et Mathews, D. H. (2007). Predicting helical coaxial stacking in RNA multibranch loops. *RNA*, 13(7):939–951.

[Ussery, 2002] Ussery, D. W. (2002). DNA structure : A-, B-and Z-DNA helix families. *Encyclopedia of Life Sciences*.

[Watson et Crick, 1953] Watson, J. D. et Crick, F. H. C. (1953). Molecular structure of nucleic acids : A structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738.

[Wilusz *et al.*, 2009] Wilusz, J. E., Sunwoo, H. et Spector, D. L. (2009). Long noncoding rnas : functional surprises from the rna world. *Genes & Development*, 23(13):1494–1504.

[Young, 2004] Young, H. P. (2004). *Strategic learning and its limits*. Oxford university press.

[Zuker, 2003] Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415.

[Zuker *et al.*, 1989] Zuker, M. *et al.* (1989). On finding all suboptimal foldings of an RNA molecule. *Science*, 244(4900):48–52.

[Zuker et Stiegler, 1981] Zuker, M. et Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Prédiction de structure tridimensionnelle de molécules d'ARN par minimisation de regret

Mots clés : ARN, structure 3D, théorie des jeux, minimisation de regret, potentiel statistique, discrétisation de l'espace.

Résumé : Les fonctions d'une molécule d'ARN dans les processus cellulaires sont très étroitement liées à sa structure tridimensionnelle. Il est donc essentiel de pouvoir prédire cette structure pour étudier sa fonction. Le repliement de l'ARN peut être vu comme un processus en deux étapes : le repliement en structure secondaire, grâce à des interactions fortes, puis le repliement en structure tridimensionnelle par des interactions tertiaires. Prédire la structure secondaire a donné lieu à de nombreuses avancées depuis plus de trente ans. Toutefois, la prédiction de la structure tridimensionnelle est un problème bien plus difficile. Nous nous intéressons ici au problème de prédiction de la structure 3D d'ARN sous la forme d'un jeu.

Nous représentons la structure secondaire de l'ARN comme un graphe : cela correspond à une modélisation à gros grain de cette structure. Cette modélisation permet de réaliser un jeu de repliement dans l'espace. Notre hypothèse consiste à voir la structure 3D comme un équilibre en théorie des jeux. Pour atteindre cet équilibre, nous utiliserons des algorithmes de minimisation de regret. Nous étudierons aussi différentes formalisations du jeu, basées sur des statistiques biologiques. L'objectif de ce travail est de développer une méthode de repliement d'ARN fonctionnant sur tous les types de molécule d'ARN et obtenant des structures similaires aux molécules réelles. Notre méthode, nommée GARN, a atteint les objectifs attendus et nous a permis d'approfondir l'impact de certains paramètres pour la prédiction de structure à gros grain des molécules.

Title : Prediction of three-dimensional structure of RNA molecules by regret minimization

Keywords : RNA, 3D structure, game theory, regret minimization, statistical potential, discrete space.

Abstract : The functions of RNA molecules in cellular processes are related very closely to its three dimensional structure. It is thus essential to predict the structure for understanding RNA functions. This folding can be seen as a two-step process: the formation of a secondary structure and the formation of three-dimensional structure. This first step is the results of strong interactions between nucleotides, and the second one is obtain by the tertiary interactions. Predicting the secondary structure is well-known and results in numerous advances since thirty years. However, predicting the three-dimensional structure is a more difficult problem due to the high number of possibility.
To overcome this problem, we decided to see the folding of the RNA structure as a game.

The secondary structure of the RNA is represented as a graph: its corresponds to a coarse-grained modeling of this structure. This modeling allows us to fold the RNA molecule in a discrete space. Our hypothesis is to understand the 3D structure like an equilibrium in game theory. To find this equilibrium, we will use regret minimization algorithms. We also study different formalizations of the game, based on biological statistics. The objective of this work is to develop a method of RNA folding which will work on all types of secondary structures and results more accurate than current approaches. Our method, called GARN, reached the expected objectives and allowed us to deepen the interesting factors for coarse-grained structure prediction on molecules.