HAL
open science

Algorithmes de graphes pour la recherche de motifs récurrents dans les structures tertiaires d'ARN

Mahassine Djelloul

To cite this version:

Mahassine Djelloul. Algorithmes de graphes pour la recherche de motifs récurrents dans les structures tertiaires d'ARN. Bio-informatique [q-bio.QM]. Université Paris Sud - Paris XI, 2009. Français. NNT : . tel-00785953

HAL Id: tel-00785953

https://theses.hal.science/tel-00785953v1

Submitted on 7 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse en cotutelle internationale

présentée par

Mahassine DJELLOUL

en vue de l'obtention du titre de

Docteur de l'Université Paris–Sud XI

Spécialité : Informatique

Docteur de l'Ecole Supérieure d'Informatique

Titre

Algorithmes de graphes pour la recherche de motifs récurrents dans les structures tertiaires d'ARN

Laboratoire de Recherche en Informatique, U.M.R. CNRS 8623,
Université Paris-Sud XI, 91405 Orsay Cedex, France
Ecole Supérieure d'Informatique, BP 68M Oued Smar, 16309 El Harrach, Algérie

Table des matières

Introduction **1**
Structure de l'ARN 1
Les motifs tertiaires de l'ARN 2
Intérêt d'analyser les motifs tertiaires 2
Définition de motif tertiaire 3
Problème d'extraction des motifs 4
Motivation de la thèse 5
Structure du document 5

1 Les motifs tertiaires **7**
1.1 Structure de l'ARN 7
1.2 Nomenclature Leontis-Westhof 11
1.2.1 Appariement de bases côté-côté (en anglais "edge-to-edge base pairing") 11
1.2.2 Empilement face-face (en anglais "face-to-face stacking") 12
1.2.3 Diagrammes 2D et réseaux d'interactions 14
1.3 Concept d'isostérie et séquence signature 14
1.4 Le problème de recherche des motifs 18

2 Modélisation informatique **21**
2.1 Cadre théorique 21
2.1.1 Quelques éléments de la théorie des graphes 21
2.1.2 Quelques éléments de la théorie de la complexité 22
2.2 Domaine d'application 27
2.2.1 Les objets : des graphes d'ARN 27
2.2.2 Le problème : calculer la similarité de deux graphes d'ARN 28
2.3 Calcul du sous-graphe commun maximum 31
2.3.1 Principe de fonctionnement des algorithmes exacts . . 32
2.3.2 Complexité théorique et performances pratiques 35

2.4 Conclusion 36

3 Extraction des motifs locaux 39
3.1 Similarité de deux motifs locaux 40
3.1.1 Définitions 41
3.1.2 Mesure de similarité 43
3.1.3 Implémentation 43
3.2 Méthode d'extraction des motifs 50
3.2.1 Module Planarisation 50
3.2.2 Module Catalogue 51
3.2.3 Module Similarité 53
3.2.4 Module Clustering 53
3.3 Résultats 57
3.4 Rna3Dmotif 60

4 Les motifs d'interaction 63
4.1 Définition de motif d'interaction 64
4.2 Similarité de deux motifs d'interaction 64
4.2.1 Définitions 64
4.2.2 Mesure de similarité 66
4.2.3 Implémentation 67
4.3 Extraction des motifs d'interaction 67
4.3.1 Données et catalogue 68
4.3.2 Module Listing 68
4.3.3 Similarité 70
4.3.4 Clustering 70
4.4 Résultats 71
4.5 Comme des pièces d'un puzzle 72

Conclusion et Perspectives 77

Annexes 81
Annexe A 81
Annexe B 83

Bibliographie 84

Introduction

Les cellules vivantes utilisent de l'énergie pour produire de la matière à partir d'un support d'information appelé code génétique contenu dans le noyau. A l'échelle moléculaire, les briques élémentaires d'information sont les gènes composés d'ADN, et les unités de matière sont les protéines composées d'acides aminés.

La synthèse des protéines à partir de l'ADN est un processus en deux temps : d'abord l'ADN est transcrit sur une molécule d'ARN messager au niveau du noyau, puis ce dernier est traduit par l'intermédiaire d'ARN non messager (ARN de transfert et ARN ribosomique) en protéine au niveau du ribosome ; l'usine à protéines de la cellule. Bien que ADN et ARN soient tous les deux des acides nucléiques, l'ADN est une molécule beaucoup plus stable que l'ARN versatile et de durée de vie temporaire.

Pendant longtemps, on a assimilé le rôle de l'ARN à celui d'un simple "drone" qui véhicule l'information génétique depuis le noyau vers le ribosome puis la convertit en protéines, mais depuis environ trois décennies, de nouveaux types d'ARN non messager ont été découverts (ARN catalytique, ARN interférent, etc.) dont les fonctions biologiques se sont révélées aussi diverses qu'indispensables à la régulation des différentes étapes de la vie cellulaire [17, 23].

Structure de l'ARN

L'ARN présente trois niveaux hiérarchiques appelés structure primaire, structure secondaire et structure tertiaire. La structure primaire est une séquence de nucléotides (A, C, G et U) qui se replie dans un premier temps en une structure intermédiaire -la structure secondaire- par la formation d'hélices Watson-Crick complémentaires. Dans cette structure secondaire jouant le rôle d'ossature de la molécule, beaucoup de nucléotides restent non appariés et forment des régions en boucles qui peuvent être vues comme les articulations de l'ossature de la structure secondaire. Les régions en boucles sont

appelées *éléments de structure secondaire*. Enfin, des liaisons de type non-Watson-Crick se forment entre différents éléments de la structure secondaire et amènent la molécule à se replier en 3D. C'est ce dernier niveau de repliement tridimensionnel, appelé structure tertiaire, qui est l'objet de notre étude.

Les motifs tertiaires de l'ARN

L'ARN est une molécule ubiquitaire qui n'est biologiquement active qu'une fois repliée dans l'espace. Chaque type d'ARN adopte une forme tridimensionnelle pré-requise à l'activation de son rôle biochimique. La compréhension des mécanismes qui président à ce repliement est encore partielle mais les recherches portant sur la structure de l'ARN ont révélé que les ARN repliés adoptent des architectures complexes pouvant être vues comme des patchworks de modules conservés appelés *motifs tertiaires* ou encore *motifs structuraux* [25, 29]. Ces motifs sont présents de manière récurrente dans les structures d'ARN et jouent un rôle qui est habituellement classé comme architectural, stabilisateur de structure, catalytique ou site d'ancrage pour protéines et ligands [56]. Cette récurrence des motifs d'ARN démontre une autonomie fonctionnelle qui fait que leur architecture 3D globale ne dépend pas du contexte structural dans lequel ils sont observés [56]. En fait, le repliement 3D du motif d'ARN est dicté essentiellement par les liaisons hydrogène entre bases et empilement des bases [49].

Intérêt d'analyser les motifs tertiaires

Le fait que la molécule d'ARN puisse être vue comme un assemblage de motifs dont la structure tridimensionnelle dépend de leur séquence s'avère d'une grande utilité. Un intérêt immédiat consiste à prédire la structure 3D à partir de la séquence uniquement. En effet, et contrairement aux séquences génomiques faciles à obtenir, la technique d'obtention de structures cristallographiques est laborieuse (une description de cette technique est donnée dans [55]). Découvrir le moyen de prédire la structure 3D d'un ARN à partir de sa séquence uniquement permet de réaliser une double économie en temps et en coût. Un autre intérêt d'analyser les motifs d'ARN est le tracé de leur conservation phylogénétique : plus un motif est conservé à travers les espèces, plus primordiale est sa fonction dans le processus de vie cellulaire. Enfin, les motifs ainsi identifiés peuvent servir de briques élémentaires dans l'ingénierie d'ARN artificiels aux propriétés spécifiques [72]. Cette technique n'a pas

encore dévoilé toutes ses promesses mais, déjà, des applications potentielles en médecine suscitent de grands espoirs.

Il apparait donc clair que plus riche est notre connaissance des propriétés structurales des motifs tertiaires, leur nombre, leurs variantes et leur fréquence d'apparition dans les structures d'ARN, meilleure sera notre compréhension du phénomène de repliement des ARN fonctionnels et plus maîtrisés seront les applications et les procédés d'ingéniérie qui en résulteront.

Définition de motif tertiaire

L'identification des motifs structuraux nécessite une définition non-ambiguë décrivant à la fois leur séquence et leur structure 3D.

En raison de la prévalence des appariements non-Watson-Crick dans les interactions tertiaires et les motifs qu'ils forment, Leontis et Westhof ont proposé une nomenclature pour l'annotation et la classification systématique des paires de bases d'ARN basée sur leur géométrie 3D [42, 41]. Cette nomenclature, connue sous le nom de *nomenclature Leontis-Westhof*, possède de multiples avantages. Elle permet (i) de décomposer un motif d'ARN en ses briques élémentaires : les paires de bases non-Watson-Crick, (ii) de dégager les règles de substitutions de bases dans les séquences d'ARN qui conservent la structure 3D des motifs, (iii) de résumer une partie essentielle de l'information de structure tridimensionnelle d'une molécule d'ARN sur un schéma bidimensionnel grâce à des symboles d'annotation faciles à mémoriser et à traiter par des programmes informatiques.

S'appuyant sur cette nomenclature, une étude menée par Lescoute et al. [45] a montré que les occurrences des motifs récurrents se partagent un noyau commun de bases et de paires de bases ayant une structure 3D similaire. Toutefois, ces occurrences diffèrent les unes des autres par l'identité des bases impliquées dans les paires de bases qui composent le motif ainsi que le nombre de bases libres ou de paires de bases qui sont insérées dans le motif sans déformer sa structure 3D globale.

Une propriété importante semble donc inhérente aux motifs tertiaires : des bases et paires de bases peuvent s'intercaler entre les paires de bases composant le motif sans en perturber significativement la structure globale.

Problème d'extraction des motifs

Les structuralistes identifient les motifs en scrutant à l'oeil nu les structures cristallographiques des molécules ("gold standard"). Malheureusement, avec la résolution continue de nouvelles molécules dont la taille peut atteindre plusieurs milliers de nucléotides comme dans l'ARN ribosomique, cette méthode manuelle atteint vite ses limites. C'est pourquoi, mettre au point des méthodes automatisant la recherche de motifs tertiaires devient un impératif auquel la bioinformatique structurale tente de répondre.

Une méthode automatique est une méthode destinée à être réalisée par un calculateur, ce qui suppose que, à la fois les objets manipulés et les actions conduites sur eux sont décrits dans un langage algorithmique compréhensible par un processeur.

Concernant l'objet d'étude, ici la structure tertiaire, plusieurs représentations en ont été proposées. Certaines utilisent à l'état brut les informations contenues dans les structures cristallographiques telles que angles de torsion du squelette [19, 68, 26] ou coordonnées cartésiennes des atomes [24, 31, 58, 61]. Les méthodes qui utilisent les angles de torsion du squelette ne sont pas tolérantes à l'insertion de bases, or l'insertion de bases ou de paires de bases entre différentes occurrences est, comme nous l'avons vu, une propriété importante dans la définition de motif structural. Les méthodes qui se basent sur les coordonnées cartésiennes des atomes sont appelées dans la littérature "méthodes géométriques" et font appel à des opérations de rotations géométriques et des calculs de métriques ou de distances intensifs. Dans l'espoir de s'affranchir de ces calculs, d'autres travaux ont eu recours à des représentations de la structure tertiaire issues de la théorie des graphes [24, 2, 37, 69] puisque le modèle de graphe est celui qui s'est avéré le mieux à même de représenter de manière naturelle la structure tertiaire et cela à des niveaux de granularité et d'abstraction différents. Toutefois, et bien que ces travaux utilisent des représentations symboliques, le processus d'identification de motifs reste hybride puisque les distances utilisées pour évaluer le degré de ressemblance entre deux occurrences sont calculées à partir des coordonnées cartésiennes des atomes.

Les méthodes hybrides d'identification de motifs appartiennent à l'une ou l'autre de deux catégories : la première cherche dans de nouvelles structures des occurrences d'un motif connu [24, 2], la seconde tente d'identifier de nouveaux motifs récurrents [37]. Dans cette dernière catégorie, le problème est rendu plus difficile puisque l'on n'a aucune idée du motif (pattern) recherché.

Hélas, un inconvénient de la démarche dans les deux catégories est que la récurrence du motif est démontrée lorsque les occurrences trouvées possèdent des modèles de graphe identiques. Or, dans la réalité, les motifs récurrents sont souvent similaires plutôt que strictement identiques.

Motivation de la thèse

Ma thèse est une contribution bioinformatique à l'identification et l'extraction automatisée de motifs tertiaires d'ARN récurrents et *a priori* inconnus. Afin de "capturer" et restituer l'information topologique contenue dans une description de structure tertiaire utilisant la nomenclature Leontis-Westhof, la méthode mise au point utilise exclusivement un modèle de graphe de la structure tertiaire dans laquelle les sommets représentent les nucléotides étiquetés par leur nom et leur numéro de séquence, et les arêtes représentent les interactions entre les nucléotides étiquetées par leurs différents types.

Transposé en théorie des graphes, le problème de recherche d'occurrences similaires appartenant à une famille de motifs récurrents (de pattern connu ou non) devient un problème d'identification de deux ou plusieurs sous-graphes similaires. Ce type de recherche fait appel au problème de calcul d'un sous-graphe commun maximum bien connu dans la littérature de la complexité algorithmique pour être NP-difficile (NP-Hard), inapproximable (APX-Hard) et ne possédant pas d'algorithme paramétré (non FPT) [32].

Me situant dans ce contexte, et en exploitant les particularités structurales des motifs tertiaires, je propose une mesure de similarité de graphe basée sur le calcul d'un sous-graphe commun maximun dont le pattern est inconnu au départ. Cette mesure de similarité permet de séparer par groupes ("clusters") les familles d'occurrences similaires représentant les motifs tertiaires potentiels. Chaque famille sera donnée par sa séquence et sa structure 3D.

Structure du document

Le manuscrit comporte quatre chapitres. Dans le chapitre 1, j'introduirai quelques notions de biologie qui permettront de se familiariser avec la terminologie relative au problème de recherche des motifs tertiaires. Le chapitre 2 sera consacré à la formulation informatique de ce problème. J'y présenterai le modèle de graphe choisi pour représenter à la fois la structure et les motifs tertiaires d'une molécule d'ARN puis je rappellerai le problème de calcul

du sous-graphe commun maximum et exposerai sa complexité algorithmique.

Dans mon travail, je m'intéresse à deux types de motifs :

1. *les motifs locaux* : ce sont des motifs insérés dans les éléments de structure secondaire. Le terme "local" veut dire "local à la structure secondaire". La plupart des motifs identifiés dans la littérature tels que le Kink-turn ou la boucle Sarcin-ricin appartiennent à cette classe de motifs. Le chapitre 3 présentera ma méthode d'identification et d'extraction automatique des motifs locaux dont les résultats ont fait l'objet d'une publication commune avec A. Denise [16].
2. *les motifs d'interaction* : ce sont des motifs longue-distance qui mettent en interaction deux ou plusieurs éléments de structure secondaire distants pour induire un repliement spatial d'une partie de la molécule. Ces motifs n'ont pas encore fait l'objet d'une recherche automatisée et le seul motif connu à ce jour appartenant à ce type est le riboze-zipper plus récemment désigné sous le nom de motif en A-mineur [57, 47]. Le chapitre 4 traitera de l'identification et classification automatiques de cette classe de motifs, ainsi que des résultats obtenus.

Chapitre 1

Les motifs tertiaires

La diversité des fonctions biologiques des ARN structurés provient de leur capacité à adopter des formes tridimensionnelles complexes. L'examen de ces structures par les experts a révélé la présence de modules conservés appelés *motifs structuraux* assemblés ou combinés de manière à mener à des architectures globulaires [4].

Le recensement de ces motifs est donc devenu une étape incontournable dans la compréhension de cette relation structure-fonction. Les nombreux travaux qui s'y sont consacré ont montré que les structures 3D de ces motifs dépendent fortement de leur séquence et forment un noyau de nucléotides caractéristique ayant des propriétés structurales distinctives.

Pour mieux comprendre le vocabulaire propre aux motifs structuraux et le problème de leur identification, commençons par voir de quoi est composé un ARN.

1.1 Structure de l'ARN

A l'échelle de la séquence, les modules unitaires d'un ARN sont les nucléotides reliés par des liens phosphodiester et orientés de l'extrémité 5' vers l'extrémité 3'. Chaque nucléotide est composé de trois entités chimiques : la base (Adénine, Cytosine, Guanine ou Uracile), le sucre (ribose) et un groupe phosphate (Figure 1.1). L'ensemble formé par la base et le sucre s'appelle nucléoside. Le nucléoside et le lien phosphodiester forment un nucléotide [70]. Les bases, et par extension les nucléotides dont elles font partie, sont, par convention, représentées par leur première lettre A, C, G et U.

Cette chaîne de nucléotides forme le *squelette sucre-phosphate* (en anglais “backbone”) de la molécule et est appelée *structure primaire*.

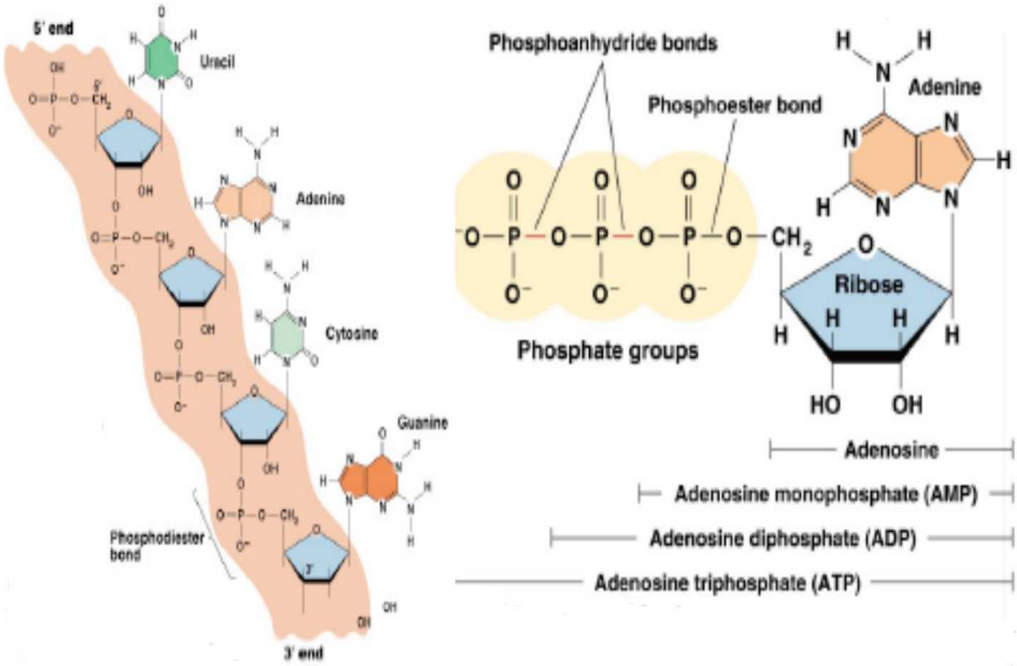


Fig. 1.1 – Composition chimique de l’ARN. Source : Pearson Education, Benjamin Cummings, Resources online, 2003.

Les polynucléotides d’ARN se replient dans le plan pour que des segments de séquences de bases complémentaires puissent se faire face et s’apparier en des paires *Watson-Crick* AU, UA, CG, et GC, ainsi que des paires “wobble” GU et UG. Deux segments d’ARN complémentaires peuvent donc s’apparier de manière antiparallèle pour former une région en duplex appelée *hélice*. La région entre les deux segments forme alors une boucle reliant les deux brins du duplex et est appelée *boucle en épingle à cheveux* ou encore *boucle terminale*. Cet ensemble d’appariements composé de régions en hélices Watson-Crick (tiges) et de régions en simple brin non-appariées (boucles) induit une topologie particulière qui s’appelle la *structure secondaire*.

Unité de mesure : La longueur des hélices d’ARN est comptée en paires de bases Watson-Crick. Cette unité est notée *pb* (en anglais “base-pairs *bp*”). Pour les ARN simple brin, on compte la longueur en nucléotides. Cette unité est notée *nt*.

Les hélices sont de courte longueur (10-15 *pb*) et sont les modules unitaires de la structure secondaire qui s'empilent de manière régulière les uns sur les autres. Les hélices elles-mêmes sont composées de briques élémentaires : les paires Watson-Crick. Les boucles reliant les hélices jouent le rôle d'articulations et portent les noms de *renflements* (en anglais "bulge"), *boucles internes*, *boucles terminales* et *boucles multiples* appelées aussi *jonctions* (Figure 1.2). Ces régions en boucles sont plus généralement appelées *éléments de structure secondaire*. Les nucléotides appartenant aux boucles sont montrés sur les représentations bidimensionnelles des structures secondaires comme n'étant pas appariés [56].

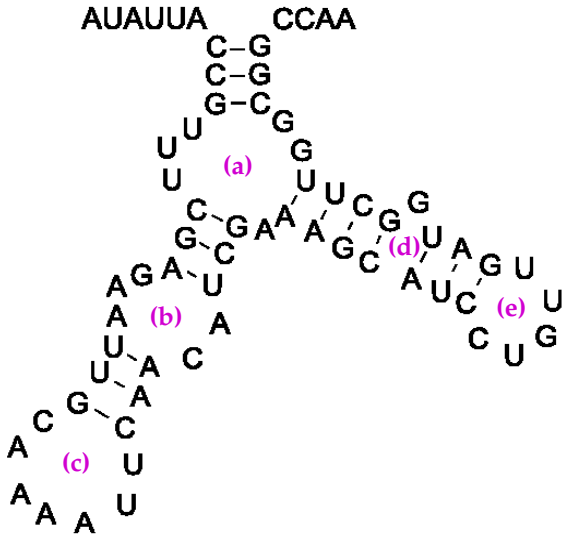


FIG. 1.2 – Représentation 2D d'un ARN. (a) : Triple jonction. (b) : boucle interne.(c) et (e) : boucles terminales. (d) : renflement. Figure extraite de [1] avec adaptation.

Pendant longtemps, le type d'appariement Watson-Crick (dit aussi *canonique*) entre deux bases est resté le mieux connu, mais la résolution de structures cristallographiques d'ARN au cours des deux dernières décennies a introduit dans la scène des masses d'informations structurales dont la taille et la richesse lancent de nouveaux défis aux chercheurs. Plus particulièrement, ces nouvelles données ont montré que les nucléotides composant les simples brins des boucles étaient souvent impliqués dans des interactions de type *non-Watson-Crick* (appelé aussi *non-canonique*) ou de type *empilement* (en anglais "stacking") formant ainsi des motifs tertiaires récurrents. Parmi les

mieux connus de ces motifs, on trouve la boucle E, la boucle Sarcin-ricin et, plus récemment, le motif Kink-turn ou encore le motif C-loop [1]. Les appariements non-Watson-Crick sont également observés dans les boucles terminales aux extrémités des hélices, à l'interface entre l'hélice et la boucle. Ils interviennent de façon très importante dans les interactions longue-distance comme dans le motif en A-mineur [49].

Historiquement, deux manières de définir un motif tertiaire ont vu le jour selon que l'on se place du point de vue du squelette sucre-phosphate ou de l'appariement des bases.

Dans la première vue, un motif est défini comme une "conformation récurrente du squelette". Toutefois, les résultats des travaux qui ont extrait des motifs en se basant sur cette définition (voir par exemple [26, 19, 68]) ont montré que ces derniers ne présentaient pas de signature caractéristique qui les relierait à la séquence. Cette limitation les rend insuffisants pour la prédiction de la structure à partir de la séquence uniquement.

La seconde vue est celle qui prend en compte les appariements de bases afin de relier la structure à la séquence. Ainsi, Leontis et Westhof définissent un motif tertiaire d'ARN comme :

> *"Un ensemble de paires de bases non-Watson-Crick isostériques ordonnées et orientées qui mènent à un repliement caractéristique du squelette sucre-phosphate."[44]*

Dans ce qui suit, j'introduis les notions permettant de comprendre cette définition.

La différence entre une interaction Watson-Crick et une interaction non-Watson-Crick se situe dans l'identité des côtés d'une base qui interviennent dans l'interaction chimique et dont l'analyse détaillée est au coeur de la classification géométrique des paires de bases d'ARN appelée nomenclature Leontis-Westhof.

[1] Pour plus d'informations sur les motifs sus-cités, le lecteur intéressé est invité à consulter les références bibliographiques fournies dans [49]

1.2 Nomenclature Leontis-Westhof

Deux nucléotides d'ARN peuvent s'apparier de différentes manières selon l'entité chimique (base, sucre ou phosphate) intervenant de chaque côté. Le type d'interaction le mieux connu implique les bases : base-base, base-sucre et base-phosphate. Les interactions base-base comportent l'appariement côté-côté et l'empilement face-face. D'autres interactions, plus rares, sont du type perpendiculaire côté-face [56]. En raison de la prédominance de leurs rôle et nombre dans l'architecture des ARN, seuls l'appariement de bases côté-côté et l'empilement seront présentés dans la suite.

1.2.1 Appariement de bases côté-côté (en anglais "edge-to-edge base pairing")

Les bases d'ARN purines (A, G) et pyrimidines (C, U) présentent trois côtés pour former des liaisons hydrogène avec d'autres bases. Ces côtés portent les noms de Watson-Crick (WC), Hoogsteen (H) et Sucre (S) comme détaillé dans la figure 1.3 (panneau de gauche, haut) pour une purine : l'Adénosine [42]. Chaque base peut donc être schématiquement représentée par un triangle dont les côtés sont étiquetés par les noms qu'ils portent (figure 1.3, panneau de gauche, bas). Une croix ou un cercle dessinés dans le coin où le côté Hoogsteen croise le côté Sucre indique l'orientation du squelette sucre-phosphate par rapport au plan de la page (5' vers 3' pour la croix et 3' vers 5' pour le cercle).

Les bases peuvent s'apparier suivant les six combinaisons des trois côtés, par exemple le côté Hoogsteen d'une base avec le côté Watson-Crick, Hoogsteen ou Sucre de l'autre base. De plus, pour chaque combinaison de côtés, les bases peuvent se positionner l'une par rapport à l'autre selon deux orientations appelées *cis* et *trans*. Dans une orientation *cis*, les liens glyscosidiques qui relient les bases à leur sucre (ribose) correspondant se trouvent du même côté de l'axe horizontal parallèle aux ponts hydrogène reliant les deux bases. Dans une orientation *trans*, les liens glyscosidiques sont situés sur des côtés opposés de cet axe (Figure 1.3, panneau de droite).

Ainsi, les six combinaisons de côtés et les deux orientations *cis* et *trans* donnent 12 familles géométriques de paires de bases (Figure 1.4).

Les paires de bases Watson-Crick canoniques appartiennent à la famille *cis* WC/WC. Les côtés Watson-Crick des bases formant des appariements non-Watson-Crick deviennent disponibles pour former des liaisons tertiaires qui

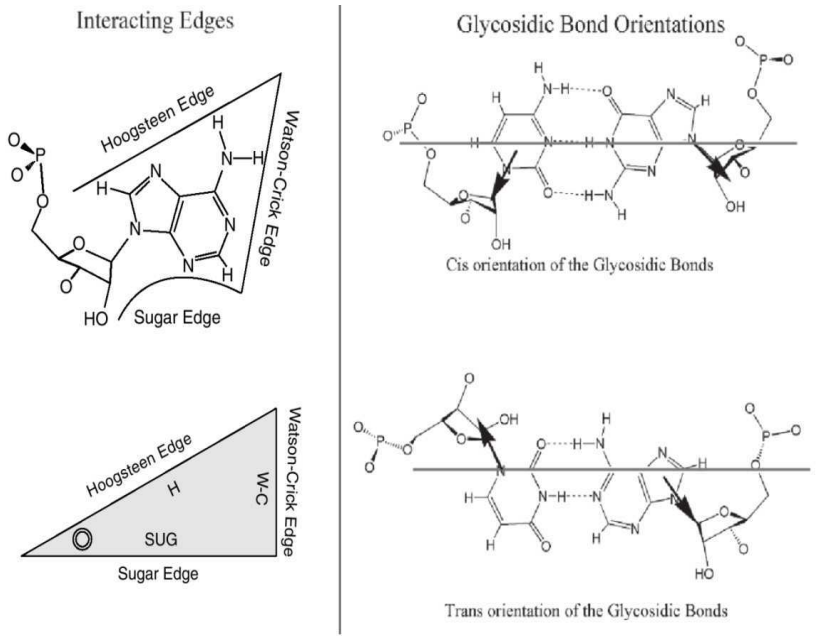


FIG. 1.3 – (Gauche) Représentation d'un nucléotide par un triangle dont les trois côtés (Watson-Crick, Hoogsteen et Sucre) peuvent former des ponts hydrogène. (Droite) Géométries *cis* et *trans* de deux bases en interaction via leurs côtés Watson-Crick [42].

stabiliseront le repliement de la structure. De plus, les bases non-appariées peuvent sortir du motif (en anglais "extrude") et s'intercaler entre d'autres bases pour former des liaisons non-canoniques ou d'empilement (voir sous-section suivante) qui ont pour rôle de stabiliser des interactions tertiaires avec d'autres régions distantes dans la structure secondaire [61].

1.2.2 Empilement face-face (en anglais "face-to-face stacking")

L'empilement des nucléotides dans les hélices joue un rôle clé dans la stabilisation des ARN repliés. Il peut être de type même-brin (en anglais "intra-strand") ou de type brins-croisés (en anglais "cross-strand" ou "interstrand"). Lorsque l'empilement fait intervenir des nucléotides non-appariés dans des boucles internes ou plus fréquemment des jonctions, il provoque une courbure entre les hélices [66]. Quand l'angle de cette courbure est de $15 \pm 15^{\circ}$, l'empilement est dit co-axial ou presque-co-axial [30].

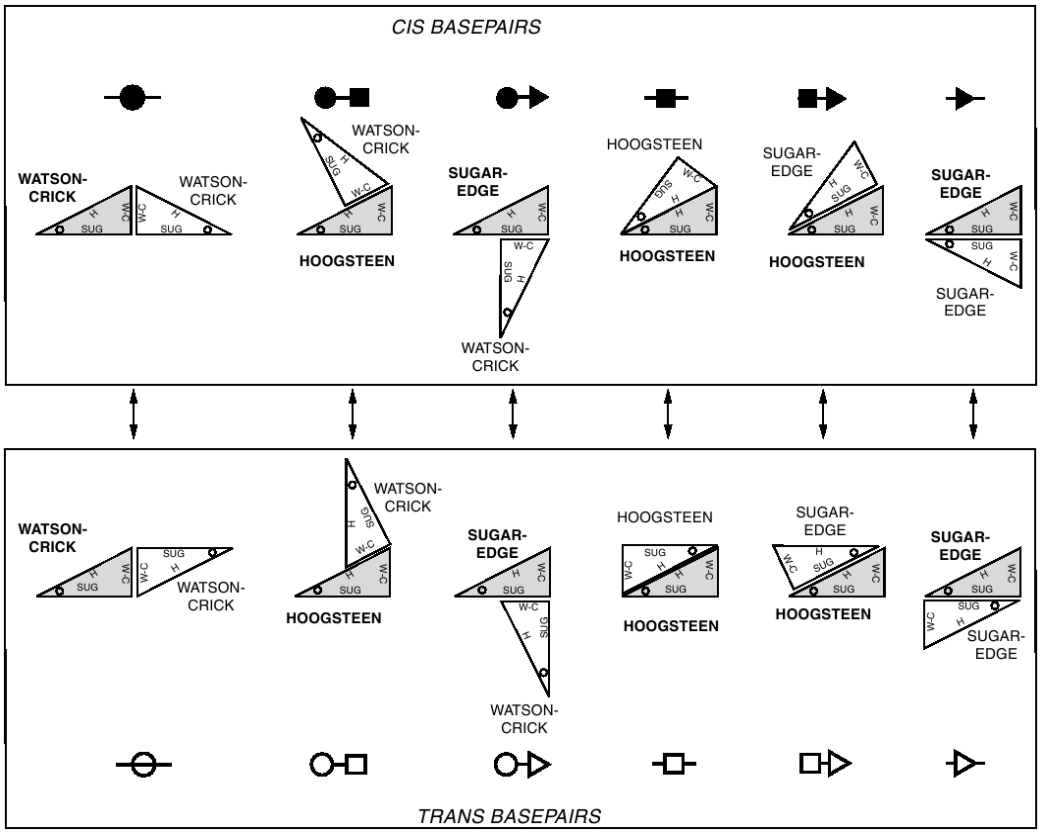


FIG. 1.4 – Douze géométries possibles de paires de bases : six *cis* (Haut) et six *trans* (Bas). Figure extraite de [43].

L'empilement dans les hélices qui implique des paires de bases Watson-Crick canoniques est une caractéristique commune de la structure secondaire et peut être prédit, de manière approximative jugée satisfaisante, par minimisation d'énergie libre [66]. En revanche, l'empilement dans les régions non-appariées est fortement influencé par la séquence [71, 53] et est notamment observé dans des motifs structuraux connus tels que le motif Sarcin-ricin (Figure 1.6).

Comme il a été procédé pour l'appariement des bases, l'empilement des bases a fait l'objet d'une classification géométrique par Sarver et collaborateurs [61]. Les faces de chaque base sont nommées selon leur orientation dans l'hélice : la face qui voit l'extrémité 3' est appelée *Face 3'* tandis que l'autre face est appelée *Face 5'*. Deux nucléotides sont considérés comme empilés s'ils sont positionnés dans des plans à peu près parallèles avec, entre autres

critères, leurs centres géométriques distants de 3 à 4.5 Å [61].

Les interactions d'empilement sont nommées selon les faces qui interagissent. Par exemple, dans une hélice régulière, chaque brin a un empilement même-brin 35, tandis qu'un empilement brins-croisés est un empilement 55 typique [61].

1.2.3 Diagrammes 2D et réseaux d'interactions

Pour faciliter l'annotation des différents types d'interactions sur des diagrammes bidimensionnels, Leontis et Westhof ont proposé d'associer à chaque type d'interaction un symbole spécifique [42]. Les nucléotides sont indiqués par leur première lettre (A, C, G et U). Aux paires Watson-Crick canoniques sont associés trois symboles différents :— pour les paires AU, = pour les paires GC, • pour les paires wobble GU. Quant aux douze familles géométriques de paires non-Watson-Crick, un ensemble de symboles noirs et blancs sont associés à chaque côté des bases appariées : un cercle pour le côté Watson-Crick, un carré pour le côté Hoogsteen et un triangle pour le côté Sucre. Quand l'orientation est *cis*, le symbole est noir, sinon, il est blanc (Figure 1.5).

La figure 1.6 montre le diagramme 2D d'un motif structural connu, la boucle Sarcin-ricin, utilisant les symboles de la nomenclature LW (où LW veut dire Leontis-Westhof).

Lescoute et Westhof [46] proposent d'exploiter la nomenclature LW pour représenter sur un dessin bidimensionnel l'information tridimensionnelle relative aux structures d'ARN. Cette nouvelle représentation est appelée *diagramme des réseaux d'interactions* (Figure 1.7). Tous les appariements Watson-Crick et non-Watson-Crick sont représentés à l'aide des symboles de la nomenclature LW [49].

Les diagrammes des réseaux d'interactions permettent de faciliter la "lecture" visuelle d'une structure cristallographique et notamment d'y repérer les motifs tertiaires en tant que petits modules similaires et récurrents (Figure 1.7).

1.3 Concept d'isostérie et séquence signature

Les structures 3D des hélices d'ARN sont régulières indépendamment de leur séquence grâce à l'isostérie des paires de bases Watson-Crick canoniques. "Isostériques" veut dire "occupant le même espace". Deux paires de bases ca-

No.	Glycosidic bond orientation	Interacting edges	Symbol	Default local strand orientation
1	*cis*	Watson–Crick/Watson–Crick		Anti-parallel
2	*trans*	Watson–Crick/Watson–Crick		Parallel
3	*cis*	Watson–Crick/Hoogsteen		Parallel
4	*trans*	Watson–Crick/Hoogsteen		Anti-parallel
5	*cis*	Watson–Crick/Sugar edge		Anti-parallel
6	*trans*	Watson–Crick/Sugar edge		Parallel
7	*cis*	Hoogsteen/Hoogsteen		Anti-parallel
8	*trans*	Hoogsteen/Hoogsteen		Parallel
9	*cis*	Hoogsteen/Sugar edge		Parallel
10	*trans*	Hoogsteen/Sugar edge		Anti-parallel
11	*cis*	Sugar edge/Sugar edge		Anti-parallel
12	*trans*	Sugar edge/Sugar edge		Parallel

FIG. 1.5 – Symboles pour annoter les 12 familles géométriques de paires de bases non-Watson-Crick. Figure extraite de [43]. Des symboles supplémentaires pour d'autres types d'interaction, dont l'empilement, peuvent être trouvés dans [42] et [56].

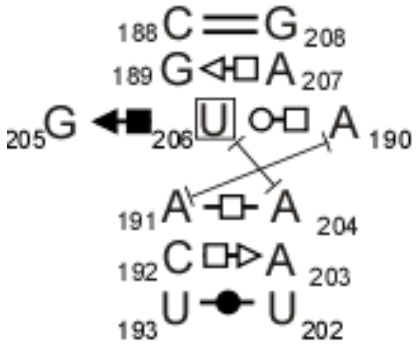


FIG. 1.6 – Diagramme 2D du motif Sarcin-ricin. Source : FR3D RNA motif library (http ://rna.bgsu.edu/FR3D/MotifLibrary/index.html)

noniques étant isostériques, elles peuvent se substituer l'une à l'autre dans une hélice sans perturber sa structure 3D. Ceci est dû au fait que l'hélice d'ARN est définie par le type d'interaction entre nucléotides et non pas l'iden-

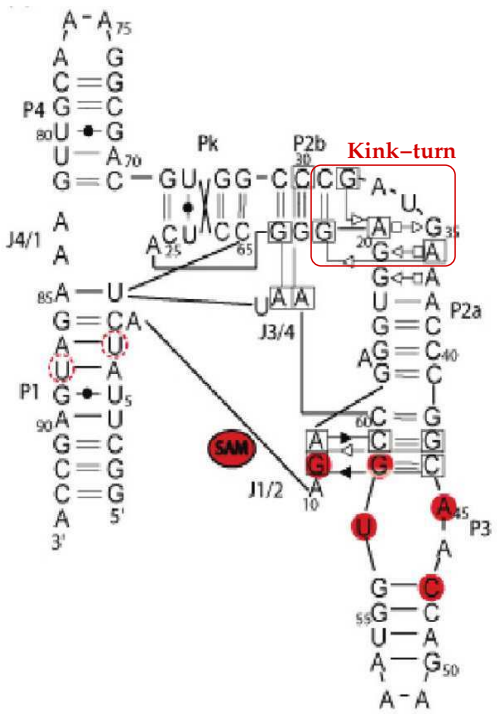


FIG. 1.7 – Diagramme du réseau d'interactions du SAM riboswitch. Un motif tertiaire, Kink-turn, est encadré en rouge. Figure extraite de [46] avec adaptation.

tité de leurs bases. Cette observation clé est à la base de l'idée de substitutions isostériques structuralement neutres. Elle s'est avérée assez féconde pour être appliquée avec succès aux paires de bases non-Watson-Crick, autrement dit, les briques élémentaires des motifs tertiaires [56].

Deux paires de bases sont dites isostériques si trois conditions sont vérifiées :(i) les distances C1'-C1' sont presque identiques, (ii) les bases correspondantes forment des ponts hydrogène entre atomes équivalents et (iii) les bases dans chaque paire sont reliées par des matrices de rotation presque identiques [64]. Pour quantifier ces trois critères, une mesure appelée *IsoDiscrepancy Index (IDI)* a été proposée par Stombaugh et al.[64]. Grâce à cette mesure, deux paires de bases sont considérées isostériques si elles possèdent un faible IDI (typiquement ≤ 3.3 Å).

Au sein d'une famille géométrique, une ou plusieurs sous-familles d'isostérie peuvent être distinguées ; chacune caractérisée par un IDI spécifique. Les paires appartenant à une sous-famille d'isostérie notée $I_{i,j}$ (où i et j repré-

sentent respectivement les index de la famille et la sous-famille d'isostérie) peuvent se substituer les unes aux autres sans perturber la structure tridimensionnelle de la paire de base.

A chacune des 12 familles géométriques, correspond une *matrice d'isostérie* (Figure 1.8). Chaque entrée de la matrice (ligne ou colonne) correspond à une base (A, C, G, U). Chaque intersection d'une ligne et d'une colonne correspond à une combinaison de bases. Si la paire de bases correspondant à cette combinaison a été observée dans les structures résolues, l'identifiant $I_{i,j}$ de la sous-famille d'isostérie à laquelle la paire considérée appartient est indiqué dans l'intersection de la ligne et la colonne correspondantes.

IM				
tWH	A	C	G	U
A	$I_{4.3}$		$I_{4.3/4.2}$	
C	$I_{4.2}$	$I_{4.1}$	$I_{4.2}$	
G			$I_{4.5}$	$I_{4.3}$
U	$I_{4.1}$		$I_{4.4}$	$I_{4.2}$

FIG. 1.8 – Matrice d'isostérie de la famille géométrique *trans* Watson-Crick/Hoogsteen. Les paires de bases AA, AG et GU appartiennent à la même sous-famille d'isostérie $I_{4,3}$. La paire AG appartient à deux sous-familles $I_{4,2}$ et $I_{4,3}$. Les cellules de la matrices colorées en gris correspondent à des combinaisons de bases non observées dans cette famillle. Figure extraite de [64].

L'isostérie des sous-familles géométriques fait que plusieurs combinaisons de séquences conduisent à un même repliement 3D du motif d'ARN. L'ensemble de ces séquences de bases qui se replient en une structure 3D similaire porte le nom de *séquence signature* du motif [44, 45, 56]. Théoriquement, la séquence signature d'un motif peut être générée de manière combinatoire à partir de toutes les paires de bases isostériques à chaque paire de base non-Watson-Crick formant le motif, mais il semblerait que d'autres contraintes structurales limitent ce nombre à quelques combinaisons uniquement [45].

La séquence signature d'un motif permet d'identifier les substitutions de bases qui conservent la structure 3D des motifs dans un alignement structural.

Cette information est d'une importance cruciale puisqu'elle facilite la prédiction de la structure tertiaire de molécules homologues sur la base de leurs séquences génomiques uniquement [45].

1.4 Le problème de recherche des motifs

Obtenir la séquence signature d'un motif récurrent nécessite d'identifier toutes ses occurrences dans les structures connues. Les biologistes le font en scrutant à l'oeil nu les structures cristallographiques. Une étude menée par Lescoute et al. [45] a montré que les occurrences des motifs récurrents sont similaires en termes de structure globale 3D mais pas nécessairement identiques à l'échelle des bases et paires de bases qui les composent. En effet, les occurrences de ces motifs ont en commun un noyau de paires de bases non-Watson-Crick isostériques empilées et ordonnées de manière à induire une structure 3D similaire, mais ces occurrences diffèrent les unes des autres par l'identité des bases formant chaque paire de base ainsi que le nombre de bases libres ou de paires de bases dont l'insertion dans le motif n'altère pas sa forme globale [61]. Ce noyau de paires de bases non-Watson-Crick isostériques commun à toutes les occurrences d'un motif récurrent et pouvant être étendu à des paires de bases Watson-Crick flanquant le motif est appelé *structure consensus* du motif.

La figure 1.9 montre quelques instances du motif Kink-turn identifiées par Lescoute et al. et représentées avec les symboles de la nomenclature LW [45].

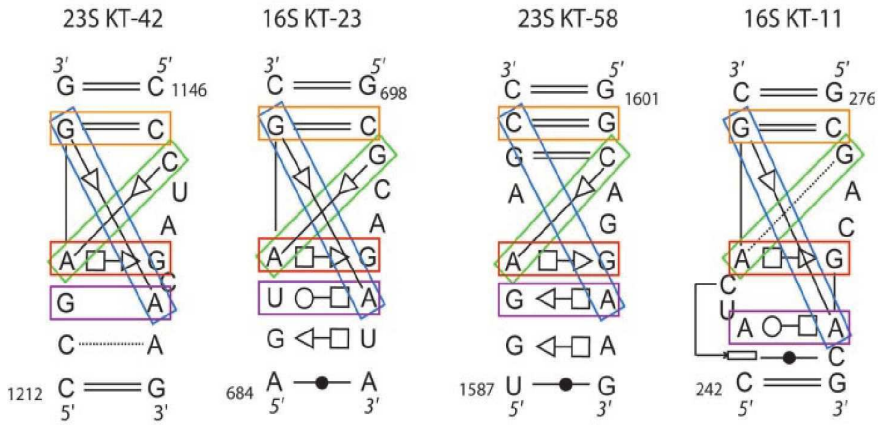


FIG. 1.9 – Quatre occurrences du motif Kink-turn identifiées à l'oeil nu dans l'ARNr 23S et 16S et similaires à celui montré dans la figure 1.7. Le consensus du motif est formé des paires de bases encadrées en couleur. Figure extraite de [45]

Dans sa version biologique, le problème de recherche de motifs tertiaires peut donc se formuler ainsi :

> **Etant donné une ou plusieurs structures tertiaires d'ARN, identifier les sous-structures récurrentes similaires (motifs) ayant un noyau commun de paires de bases non-Watson-Crick isostériques.**

Les experts identifient les motifs à l'oeil nu. L'automatisation de cette étape non seulement trouvera les motifs de manière exhaustive, mais accordera plus de temps aux experts pour se consacrer à des tâches exigeant une expertise exclusivement humaine.

Une méthode automatique destinée à être exécutée par un calculateur travaille sur un modèle de la structure 3D. Justement, il existe un modèle qui s'est avéré idéal pour représenter la topologie d'une structure tertiaire décrite selon la nomenclature LW. Ce modèle est celui de *graphe*.

Dans le chapitre suivant, je présente le modèle de graphe choisi pour représenter une structure tertiaire et ses motifs puis j'expose le pendant informatique du problème de recherche des motifs tertiaires.

Chapitre 2

Modélisation informatique

La section 1 donne les notions théoriques qui permettront de comprendre le vocabulaire informatique employé dans ce chapitre. Dans la section 2, je reviens à notre domaine d'application en définissant notre objet d'étude le modèle de graphe d'ARN puis, je formule le pendant informatique du problème de recherche de motifs similaires. La résolution de ce problème fait appel au calcul d'un sous-graphe commun maximum. Aussi, je consacre la section 3 à un état de l'art des principales approches utilisées dans les algorithmes exacts de calcul du sous-graphe commun maximum.

2.1 Cadre théorique

2.1.1 Quelques éléments de la théorie des graphes

Tout système qui consiste en un ensemble discret d'états ou de sites appelés noeuds ou sommets entre lesquels des liens appelés arêtes ou arcs traduisent une certaine relation, peut être modélisé par un graphe.

Un **graphe** $G = (V, E)$ est donné par un ensemble V de sommets ("vertices") et un ensemble d'arêtes ("edges") $E \subseteq V \times V$. L'**ordre** d'un graphe est le nombre de ses sommets, et sa **taille** est le nombre de ses arêtes. Deux sommets reliés par une arête sont dits **voisins**, et une arête reliant deux sommets v et w est dite **incidente** à v et w et est notée (v, w).

Les arêtes peuvent être **orientées** ou non. Une arête orientée s'appelle un **arc**. Une arête non orientée est dite **symétrique**. Dans la suite, j'utiliserai les termes d'**arête orientée** et **arête symétrique**.

Un graphe **étiqueté** est un graphe où les sommets et/ou les arêtes ont des attributs supplémentaires comme un nom, une couleur ou un poids.

Un graphe **complet**, dit aussi **clique**, est un graphe où tous les sommets sont reliés deux à deux.

Un graphe $H = (W, F)$ est un **sous-graphe** d'un graphe $G = (V, E)$ si $W \subseteq V$ et $F \subseteq E$. Si de plus F contient *toutes* les arêtes de E reliant deux quelconques des sommmets de W alors H est appelé **sous-graphe induit** par W (Figure 2.1).

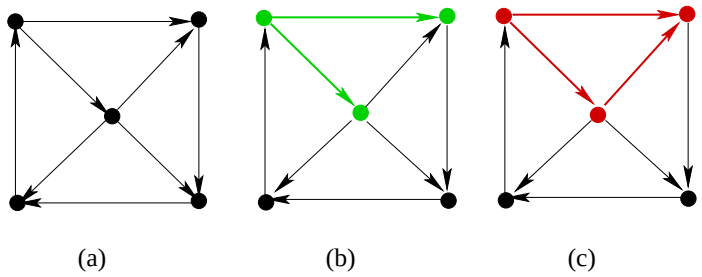


FIG. 2.1 – (a) Un graphe orienté d'ordre 5 et de taille 8. (b) En vert, un sous-graphe d'ordre 3 et de taille 2. (c) En rouge, un sous-graphe induit d'ordre 3 et de taille 3.

2.1.2 Quelques éléments de la théorie de la complexité

L'optimisation combinatoire dans les graphes est un domaine qui s'intéresse à la recherche dans de tels systèmes de structures optimales étant donné un ensemble de paramètres à optimiser. Des "casse-tête" célèbres comme le problème du voyageur de commerce ou le problème de planification des emplois du temps ont suscité beaucoup de passions des chercheurs. Il ne s'agit pas simplement de trouver mais de trouver efficacement. A lui seul, ce mot peut résumer l'idée qui préside les fondements de la théorie de la complexité.

Depuis que Alan M. Turing a défini la notion d'algorithme il y a plus d'un demi-siècle, on a pu s'accorder sur ce qu'on appelle temps de calcul d'un algorithme et, par conséquence, un algorithme efficace. Une caractérisation importante des problèmes de décision a vu le jour : la NP-complétude. Un **problème de décision** est un problème pour lequel on veut savoir si la réponse est "oui" ou "non". Par exemple, étant donnés un graphe et un entier

k, existe-t-il une clique dont le nombre de sommets serait $\geq k$?

Un codage de données est dit **raisonnable** si l'encombrement mémoire nécessaire pour stocker un nombre positif N est égal à $\lceil log(N) \rceil$ où $\lceil x \rceil$ désigne le plus petit entier supérieur ou égal à x. Ceci, quelle que soit la base choisie pour le logarithme (habituellement, on prend le logarithme en base 2 ; base de fonctionnement des ordinateurs).

Un algorithme est dit (de temps) **polynomial** si le nombre d'opérations élémentaires utilisées par l'algorithme sur une donnée dont le codage raisonnable est de taille n, est une fonction polynomiale en n. Un algorithme est considéré comme efficace si l'ordre de son nombre d'opérations est au plus celui de l'ordre d'un polynôme en n.

Habituellement, on se satisfait dans le calcul de la complexité algorithmique d'une approximation asymptotique de l'encombrement mémoire. Ainsi, pour stocker un nombre N, on se contentera de dire que l'espace requis est en $O(logN)$. Pour d'autres définitions, voir par exemple [73].

Un **problème** est dit **polynomial** s'il existe un algorithme polynomial pour le résoudre, c'est à dire un algorithme polynomial en la taille d'une instance du problème.

Remarque : La complexité des temps d'exécution est souvent donnée *au pire*. Elle correspond au temps maximal au bout duquel on est certain que l'algorithme se termine et cela quelle que soit la taille du problème entrée.

Un problème de décision est **dans la classe NP** (Non deterministic Polynomial) s'il admet un algorithme qui, lorsqu'on lui fournit une instance du problème et une solution pour cette instance (appelée certificat), il peut tester en un temps polynomial en la taille des données, si la réponse pour cette instance est "oui".

Une caractéristique notable des problèmes de NP est que leur solution est difficile à trouver, mais si elle est trouvée, elle est facile à vérifier.

La classe NP contient la classe P des problèmes polynomiaux. En effet, étant donné que le problème lui même peut être résolu en temps polynomial, il n'y a plus besoin de vérifier, pour une solution donnée, si la réponse est "oui".

La question la plus célèbre de la théorie de la complexité et qui résiste à tous

les chercheurs est : “est-ce que P et NP sont différents ?” En effet, si pour un problème dans NP on ne trouve pas *actuellement* un algorithme polynomial qui le résout, qu’en est-il dans l’avenir ? Autrement dit, on ne peut affirmer que P et NP sont différents que si, pour *au moins* un problème de NP on arrive à un moyen de prouver qu’il n’existera jamais d’algorithme polynomial pour le résoudre.

Notons C la classe des problèmes de NP pour lesquels on ne connaît pas encore d’algorithme polynomial. Un problème π de C est **NP-complet** si, pour tout autre problème de C, il existe une réduction polynomiale qui le “ramène” à π. La NP-complétude apparaît ainsi comme une mise en évidence, parmi les problèmes de C, de ceux qui sont les plus difficiles à traiter. Plus précisément, l’idée de la réduction polynomiale “réduit” la recherche d’algorithme polynomial à un seul (au choix) des problèmes montrés comme étant NP-complets. Si un jour on trouve un algorithme polynomial pour un seul problème NP-complet, alors on aura prouvé que P=NP.

Un problème H est **NP-difficile** si et seulement si il existe un problème D de décision qui se réduit en temps polynomial à H. Ainsi, si un jour on découvre un algorithme polynomial qui résout D, alors H peut aussi être résolu en temps polynomial par ajout d’une simple étape : le temps d’exécution de la réduction. Un problème NP-difficile est donc un problème qui est au moins aussi difficile qu’un problème NP-complet (Figure 2.2).

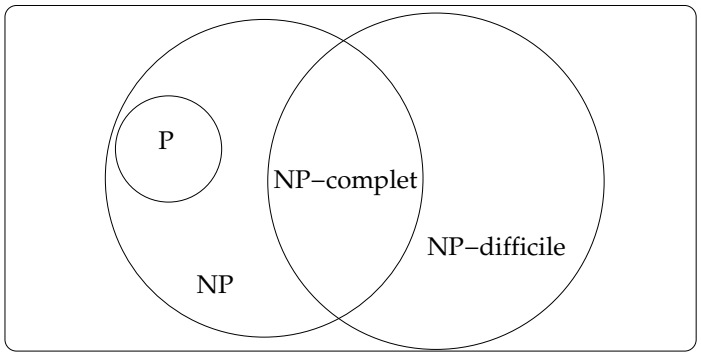


Fig. 2.2 – Classes P, NP, NP-complet et NP-difficile.

On étend cette définition en disant qu’un **problème d’optimisation** est NP-difficile si le problème de décision correspondant est NP-complet. Par exemple, l’énoncé : “Trouver une clique ayant un nombre maximum de sommets” est la version problème d’optimisation du problème de décision dont

l'énoncé est : "étant donnés un graphe et un entier k, existe-t-il une clique dont le nombre de sommets serait $\geq k$?"

Un exemple de réduction de ce problème de décision à son problème d'optimisation serait le suivant. Supposons que notre graphe ait 1000 sommets, et supposons que nous ayons un algorithme polynomial qui résout le problème de décision. Alors on pose la question : "existe-t-il une clique d'ordre 500 ?" Si la réponse est "oui" alors l'ordre de la clique maximum est certainement compris entre 500 et 1000. On pose alors la question : "existe-t-il une clique d'ordre 750 ?" Si la réponse est "oui" alors l'ordre de la clique maximum est certainement compris entre 750 et 1000. On continue ainsi de suite par dichotomie jusqu'à trouver la valeur exacte de l'ordre de la clique maximum.

Approche par approximation

Puisque malgré les efforts les plus tenaces la question ne cède pas, on a abordé le problème d'une autre façon. Pour un problème NP-difficile, on cherche à mettre au point des algorithmes polynomiaux d'approximation, c'est à dire des algorithmes qui tournent en un temps polynomial en la taille des données pour fournir une solution proche de l'optimum.

Un algorithme d'approximation pour un problème NP-difficile est un algorithme qui, étant donnée *une* constante fixée ε, fournit une solution qui est au pire ε fois pire que l'optimum. Autrement dit, s'il s'agit d'un problème de maximisation (resp. de minimisation), la solution fournie est au pire ε fois plus petite (resp. plus grande) que la solution optimale. Ici, ε est appelé rapport d'approximation.

Un algorithme d'approximation qui fournirait une solution en temps polynomial pour *toute* constante fixée ε est appelé schéma d'approximation polynomial ("**polynomial time approximation scheme (PTAS)**"). Le temps d'exécution d'un PTAS doit être polynomial en la taille n du problème pour tout ε fixé. Ainsi, un algorithme d'approximation tournant en $O(n^{1/\varepsilon})$ est considéré comme un PTAS même si le temps croit rapidement pour des valeurs décroissantes de ε. Un meillleur temps d'exécution pour cet algorithme est un temps polynomial à la fois en la taille du problème et en $1/\varepsilon$. Un PTAS avec un temps d'exécution ayant ces propriétés est appelé schéma d'approximation totalement polynomial ("**fully polynomial approximation scheme (FPTAS)**").

Puisque tous les problèmes NP-complets sont réductibles les uns aux autres, il

était naturel de penser qu'ils possèdent la même approximabilité. Autrement dit, un algorithme d'approximation applicable à un problème NP-complet A est aussi valable pour un autre problème NP-complet B. Ceci n'est cependant pas vrai pour deux raisons. La première est que les réductions NP ne préservent pas toujours la transformation de la fonction objectif du problème A vers la fonction objectif du problème B. La seconde raison est que même lorsque les fonctions objectifs sont préservées, l'approximabilité, elle, ne l'est pas nécessairement [33].

Comme les réductions NP ne préservent pas systématiquement l'approximabilité, on a dû introduire un autre type "plus fort" de réduction [33]. De telles réductions permettent non seulement de transformer chaque instance d'un problème A en une instance d'un problème B, mais aussi de transformer chaque solution approchée de cette instance de B en une solution approchée de l'instance originale de A. La qualité de cette solution dépend à la fois de la réduction et de l'algorithme d'approximation [33].

Comme il est de coutume en théorie de la complexité, on a voulu catégoriser les problèmes NP-complets selon leur approximabilité. Des classes d'approximation ont été introduites pour regrouper ensemble les problèmes NP-complets qui ont un même comportement en termes d'approximabilité.

Soit FPTAS la classe des problèmes d'optimisation pouvant être approximés par un schéma d'approximation totalement polynomial, et soit PTAS la classe des problèmes d'optimisation pouvant être approximés par un schéma d'approximation polynomial. Soit aussi APX la classe des problèmes pouvant être approximés par un *certain* rapport constant $c \geq 1$. Ainsi, un problème **APX-difficile** est un problème qui ne possède pas de schéma d'approximation polynomial. L'appartenance d'un problème étudié à cette classe est généralement une mauvaise nouvelle.

La hiérarchie des classes d'approximation est la suivante :

$$P \subseteq FPTAS \subseteq PTAS \subseteq APX \subseteq NP$$

Complexité paramétrée

A la fin des années 1990, un autre concept est introduit pour aborder la NP-complétude, appelé *complexité paramétrée* [18]. Pour beaucoup de problèmes NP-complets, l'explosion combinatoire est souvent due à une petite partie du problème appelée **paramètre**. Ce paramètre est souvent un entier

petit dans la pratique (même si la taille du problème est grande). Les temps d'exécution d'algorithmes simples peuvent être exponentiels en le paramètre mais polynomiaux en la taille du problème.

Dans la complexité paramétrée, on cherche des algorithmes de résolution exacte qui tournent en temps $f(k)n^{\beta}$, où k est un entier (le paramètre),f une fonction arbitraire ne dépendant que de k, n la taille non paramétrée du problème et β une constante. Schématiquement, on cherche à "contenir" l'explosion combinatoire du problème dans la partie $f(k)$.

De tels problèmes sont dits "**fixed-parameter tractable**" et leur ensemble forme la classe FPT. Un exemple de problème FPT est celui de la couverture des arêtes d'un graphe par les sommets ("k-Vertex Cover problem") dont l'énoncé est le suivant : "Etant donné un graphe $G = (V, E)$ d'ordre n et un entier k, trouver une couverture des arêtes par les sommets ("vertex cover") de cardinalité k, i.e. un sous-ensemble V' ayant k sommets tel que chaque arête de G est incidente à au moins un sommet de V'."
Ce problème possède un algorithme paramétré qui le résout en un temps $O(kn + 1.286^k)$ [11].

La notion de "fixed-parameter tractability" a été introduite pour appréhender des problèmes NP-difficiles paramétrés, et on a pu montrer que nombre de ces problèmes sont dans FPT. Toutefois, certains problèmes paramétrés ne possédent pas d'algorithmes paramétrés. Ces problèmes sont appelés **"fixed-parameter intractable"**. Un grand nombre de problèmes paramétrés ont été identifiés comme fixed-parameter intractable. Un de ces problèmes est celui de la clique maximum [32].

2.2 Domaine d'application

2.2.1 Les objets : des graphes d'ARN

Dans la suite, je modélise une structure tertiaire par un graphe, appelé **graphe d'ARN** où les sommets représentent les nucléotides étiquetés par leur base et leur numéro dans la séquence, et les arêtes représentent les interactions entre les nucléotides. Ces arêtes étiquetées et orientées sont de quatre types (Table 2.1).

La figure 2.3 illustre un motif Sarcin-ricin et son modèle de graphe.

Type d'arête	Etiquettes	Orientation
Squelette sucre-phosphate	{a}	de l'extrémité 5′ vers l'extrémité 3′
Canonique (AU, GC, GU)	{n}	symétrique
Non-canonique	{b,c, ...m}	selon la règle WC > H > S si les côtés des bases reliées sont différents, symétrique dans le cas contraire
Empilement	{s}	symétrique

TAB. 2.1 – Etiquetage d'un graphe d'ARN.

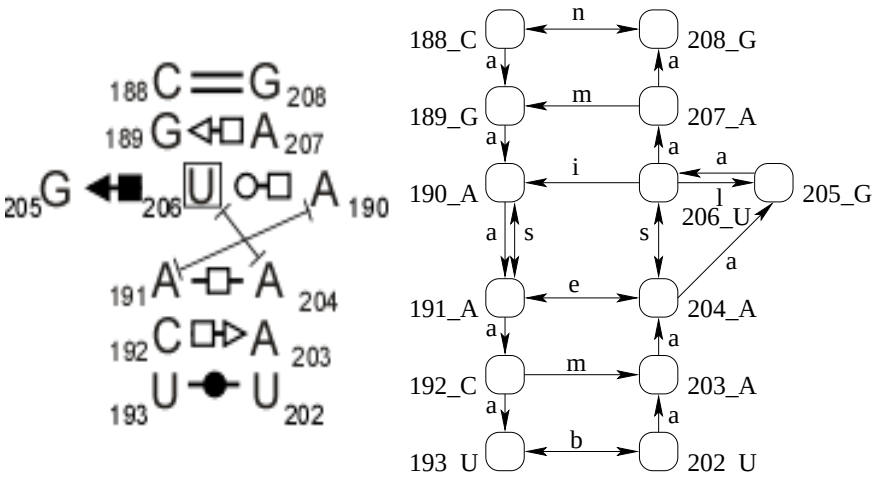


FIG. 2.3 – Graphe d'ARN du motif Sarcin-ricin.

2.2.2 Le problème : calculer la similarité de deux graphes d'ARN

Dans sa version informatique, le problème de recherche de motifs tertiaires défini en section 1.4 se formule ainsi :

> **Etant donné un ou plusieurs graphes d'ARN (structures tertiaires), identifier les sous-graphes similaires (motifs) ayant un ensemble commun d'arêtes non-canoniques.**

La définition exacte de la similarité de deux sous-graphes d'ARN sera donnée dans le chapitre 3. Dans ce qui suit, je présente quelques notions introduc-

tives sur la similarité de graphes.

La similarité entre deux objets est définie en termes de correspondances, la dissimilarité en termes de différences (en anglais "matches/mismatches"). La comparaison peut porter sur des objets entiers ou seulement des régions locales de ces objets. Dans le premier cas on parle de similarité globale tandis que dans le second cas il s'agit de similarité locale [52].

Lorsque les objets comparés sont des graphes, le problème de détermination de la similarité est connu sous le nom de *graph-matching problem* et se formule ainsi :

Etant donné deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$, trouver une bijection $M \subseteq V_1 \times V_2$ telle que :

$$\forall u, v \in V_1, (u, v) \in E_1 \Leftrightarrow (M(u), M(v)) \in E_2.$$

Note : Si G_1 et G_2 sont étiquetés, alors M doit également préserver la correspondance des étiquettes des sommets et des arêtes.

Lorsque M existe, elle est appelée un **isomorphisme** de G_1 vers G_2, et le problème correspondant est du type *exact graph-matching*.

Lorsqu'un isomorphisme est impossible à trouver, on cherche non plus une correspondance exacte entre les sommets, mais la **meilleure** correspondance possible entre eux. Cette classe de problèmes est appelée *inexact graph-matching* [5].

La similarité est quantifiée par une mesure : plus grande est cette mesure, plus similaires sont les objets comparés (ici des graphes). Inversement, plus grande est la mesure de dissimilarité, plus les objets comparés sont différents.

Lorsqu'une mesure de dissimilarité vérifie les propriétés d'une métrique, elle est appelée **distance**[1]. Ces propriétés sont :

- $distance(G_1, G_2) \geq 0$ (non-négativité)
- $distance(G_1, G_2) = 0 \Longrightarrow G_1$ isomorphe à G_2
- $distance(G_1, G_2) = distance(G_2, G_1)$ (Symétrie)
- $distance(G_1, G_3) \leq distance(G_1, G_2) + distance(G_2, G_3)$ (Inégalité triangulaire)

[1]Plusieurs mesures de distance et de similarité sont décrites dans [15].

Une approche classique pour comparer deux graphes est basée sur l'idée de calcul de leur distance d'édition, autrement dit, le coût minimal pour transformer un graphe en un autre en appliquant des opérations d'édition élémentaires (voir par exemple [8]). Or, définir les opérations d'édition ainsi que leurs coûts associés non seulement dépend fortement du domaine d'application mais est parfois problématique dans la mesure où deux graphes considérés comme similaires par une fonction de coût peuvent s'avérer dissimilaires selon une autre [27].

Dans une approche alternative pour mesurer la distance entre deux graphes, on cherche une sous-structure commune aux deux graphes qui satisfait certaines propriétés et qui est typiquement de cardinalité maximale. Par exemple, Bunke & Shearer ont développé une mesure de distance de graphe basée sur le calcul d'un sous-graphe commun maximum [10]. Cette approche possède l'avantage de ne pas nécessiter le calcul explicite d'une fonction de coût. On notera qu'il a été démontré que, sous certaines conditions concernant la fonction de coût, le calcul d'un sous-graphe commun maximum est équivalent au calcul d'une distance d'édition [9].

Pour revenir à notre contexte des motifs tertaires similaires, il est clair que la seconde approche est la plus appropriée puisque nous avons besoin de calculer le sous-graphe commun maximum de deux sous-graphes d'ARN (correspondant à deux occurrences d'un motif potentiel) car il correspondra, au cas où il existerait, à la structure consensus de la famille recherchée.

Nous avons vu au chapitre précédent que deux instances d'un motif récurrent peuvent être ou bien totalement similaires (i.e. leurs graphes sont isomorphes) ou bien partiellement similaires (i.e. leurs graphes possèdent un sous-graphe commun de taille non nulle). Schématiquement, ce sous-graphe commun maximum peut être vu comme l'intersection des deux graphes comparés. Lorsque cette intersection couvre la totalité des deux graphes, la correspondance trouvée est un isomorphisme. Ainsi le problème d'isomorphisme de graphes peut être considéré comme un cas particulier du problème de calcul du sous-graphe commun maximum. On remarquera, dans ce cas, que la (mesure de) similarité de deux graphes est proportionnelle à la taille de leur sous-graphe commun maximum. Elle est maximale lorsque les deux graphes sont isomorphes.

2.3 Calcul du sous-graphe commun maximum

Un graphe G_{12} est un sous-graphe induit commun à deux graphes G_1 et G_2 s'il est isomorphe à deux sous-graphes induits de G_1 et G_2. Un sous-graphe induit commun maximum, en anglais *maximum common induced subgraph (MCIS)* est un sous-graphe induit commun ayant le plus grand nombre de sommets. Une notion proche du MCIS est celle du *maximum common edge subgraph (MCES)*. Un MCES est un sous-graphe commun ayant le plus grand nombre d'arêtes [60].

Dans une recherche de sous-graphe commun maximum, une distinction supplémentaire peut être faite entre le cas connexe et le cas non-connexe. Un sous-graphe commun est connexe si, pour chaque paire de ses sommets, il existe un chemin les reliant. On dit qu'il forme une seule composante connexe. Par opposition, un sous-graphe commun est non-connexe s'il est composé de deux ou plusieurs composantes connexes [60].

Garey & Johnson ont montré que le problème de calcul d'un sous-graphe commun maximum de deux graphes est NP-difficile dans le cas général [22]. Par réduction au problème de la clique maximum, Kann a également montré qu'il est APX-difficile [34], ce qui veut dire qu'il ne possède pas de schéma d'approximation polynomial (PTAS). Enfin, et toujours par réduction au problème de la clique maximum, on a montré [32] que le problème du sous-graphe commun maximum est fixed-parameter intractable, autrement dit, il ne possède pas d'algorithme paramétré. Ce problème donc est l'un des plus difficiles à résoudre du point de vue de la complexité algorithmique.

Pour avoir un ordre de grandeur de la complexité de calcul d'un sous-graphe commun induit d'ordre k, l'idée simple serait d'énumérer tous les sous-graphes induits d'ordre k du premier graphe, énumérer tous les sous-graphes induits d'ordre k du second graphe, puis comparer chaque paire de sous-graphes induits en testant toutes les permutations possibles des k sommets. Ce qui donnerait, pour deux graphes d'ordres n_1 et n_2 respectivement :

$$\frac{n_1!}{(n_1-k)!k!} \cdot \frac{n_2!}{(n_2-k)!k!} \cdot k!$$

comparaisons à effectuer, autrement dit un nombre exponentiel pour des valeurs non triviales de n_1, n_2 et k [60].

Malgré cette complexité algorithmique, on a cherché à mettre au point des heuristiques utilisables en pratique. Deux grandes classes de méthodes ont

vu le jour. La première classe cherche une solution exacte au problème du sous-graphe commun maximum, la seconde calcule une solution approchée. Dans la suite, je ne présenterai que la première classe vu que notre problème d'identification de motifs tertiaires requiert une solution exacte. L'obtention de cette solution exacte est possible car l'ordre de nos sous-graphes d'ARN est inférieur à 40 sommets. Pour une revue des méthodes approchées, le lecteur intéressé est invité à consulter par exemple [60, 13].

2.3.1 Principe de fonctionnement des algorithmes exacts

Les algorithmes exacts trouvent un sous-graphe commun maximun (MCIS ou MCES) en construisant un arbre de recherche énumérant tous les sous-graphes communs possibles et choisissant le (ou les) plus grand(s). Deux principales approches sont connues. La première consiste à transformer le problème de recherche du sous-graphe commun maximum en un problème de recherche d'une clique maximum dans le *graphe des compatibilités* des deux graphes comparés. La seconde approche construit un espace d'états par énumération exhaustive de tous les sous-graphes communs possibles.

a) Approche par réduction au problème de la clique maximum

Etant donné deux graphes à comparer, on commence par construire leur *graphe des compatibilités* (appelé aussi *graphe des associations* ou encore *graphe produit*). Ce graphe des compatibilités possède la propriété suivante : un sous-graphe commun aux deux graphes comparés correspond à une clique dans le graphe des compatibilités, et donc trouver un sous-graphe commun maximum revient à trouver une clique maximum dans le graphe des compatibilités (Figure 2.4).

Définition 1 Le **graphe des compatibilités** de deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ est un graphe $H = (W, F)$ tel que $W = V_1 \times V_2$ et une arête relie deux sommets $\{u_1, u_2\}$ et $\{v_1, v_2\}$ dans W si :

$$((u_1, v_1) \in E_1 \text{ et } (u_2, v_2) \in E_2) \text{ ou bien } ((u_1, v_1) \notin E_1 \text{ et } (u_2, v_2) \notin E_2)$$

Bien que la recherche d'une clique maximum dans un graphe soit également un problème NP-difficile [22], plusieurs algorithmes rapides en pratique ont été proposés pour le résoudre. Des revues commentées de ces algorithmes sont fournies dans [60, 13].

L'idée de base d'un algorithme de recherche d'une clique maximum est la suivante. On énumère toutes les cliques maximales (i.e. cliques ne pouvant

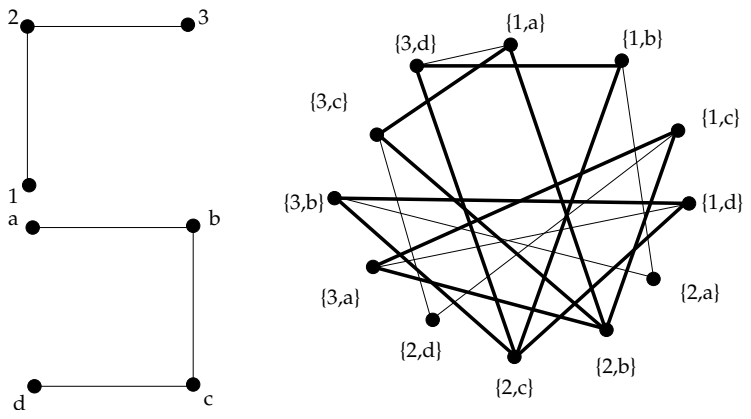


FIG. 2.4 – Graphe des compatibilités de deux graphes connexes. En gras, quatre cliques maximum ($\{1,a\}$, $\{2,b\}$, $\{3,c\}$), ($\{1,b\}$, $\{2,c\}$, $\{3,d\}$), ($\{1,c\}$, $\{2,b\}$, $\{3,a\}$), ($\{1,d\}$, $\{2,c\}$, $\{3,b\}$) correspondent à quatre MCIS possibles.

plus être étendues), puis on choisit celle(s) ayant le plus grand nombre de sommets.

Considérons le problème de trouver toutes les cliques maximales dans un graphe $G = (V, E)$. Une clique courante C peut être étendue s'il existe un sommet de V qui n'est pas encore dans C mais qui est relié à tous les sommets de C. Appelons P l'ensemble de ces sommets candidats. Toutes les extensions possibles de C avec des sommets de P peuvent être obtenues en prenant un sommet à la fois dans P et l'ajoutant à C, créant ainsi un nouvel ensemble P à partir de l'ancien P duquel on supprime tous les sommets qui ne sont pas voisins à celui qui vient d'être ajouté à C. Cette procédure est appelée de manière récursive sur les nouveaux ensembles C et P.

Pour éviter d'énumérer des cliques contenues dans d'autres cliques préalablement détectées, on définit un ensemble I contenant les sommets de V qui ont déjà servi à étendre la clique courante C. Toutes les extensions possibles de la clique peuvent être obtenues en choisissant un sommet candidat à la fois qui est dans P mais pas dans I et l'ajoutant à C. Cette procédure est appelée récursivement sur les nouveaux ensembles C, P et I obtenus à partir des anciens ensembles par suppression des sommets qui ne sont pas voisins à celui qui vient d'être choisi. Grâce à cette procédure, une clique est maximale si P et I sont tous les deux vides.

Cet algorithme peut être transformé en un algorithme branch-and-bound

pour trouver une clique maximum. L'idée est de "prédire" à chaque étape si la taille de la clique courante peut être plus grande que la plus grande clique déjà trouvée en vérifiant, parmi d'autres critères d'élagage, si le nombre de sommets candidats restants est suffisant pour obtenir une clique plus large que la clique maximum courante.

Cette description est basée sur celle donnée dans [67] de l'algorithme de Bron & Kerbosh datant de 1973 [7]. D'autres améliorations en vue d'accélération de la recherche y ont été apportées depuis. En 1999, Durand et al. décrivent un algorithme de recherche d'un MCES de deux graphes représentant des structures chimiques par réduction au problème de la recherche d'une clique dans le graphe des compatibilités [20]. Une amélioration de cet algorithme qui était déjà une adaptation de l'algorithme de Bron & Kerbosh a été implémentée dans [12] à des fins de comparaison de performances (voir section suivante).

b) Approche par énumération des sous-graphes communs

Lorsque les graphes comparés sont peu denses, le graphe des compatibilités devient dense et la recherche d'une clique maximum devient coûteuse. Une approche alternative pour ce type de graphes consiste à construire un arbre de recherche qui énumère tous les sous-graphes communs possibles.

L'algorithme le plus notable de cette catégorie est celui dû à McGregor [50]. Une variante de cet algorithme a été implémentée dans [12] à des fins de comparaison de performances (voir section suivante). Je donne, dans ce qui suit, une description du principe de fonctionnement de cet algorithme. Supposons qu'on veuille trouver un sous-graphe commun induit maximum des deux graphes G_1 et G_2 de la figure 2.5 (d'ordres n_1 et n_2 respectivement et tels que $n_1 \leq n_2$).

On commence par construire un arbre de recherche. Les variables du premier niveau de cet arbre sont les sommets du plus petit graphe. Pour chaque variable $v_1 \in G_1$, on teste si elle est compatible avec un sommet $v_2 \in G_2$ (il y en a n_2 en tout). Lorsqu'une association de sommets n'est pas valide (i.e. les sous-graphes communs correspondants ne sont pas isomorphes), la branche est élaguée. Au deuxième niveau, on essaie d'étendre les associations de sommets valides du niveau précédant, autrement dit on va tester si les sommets non encore associés du graphe G_1 (en tout (n_1-1)) peuvent être associés avec les sommets restants de G_2 (en tout (n_2-1)). Une fois une association de sommets explorée, on remonte au niveau précédant ("backtrack") pour resto-

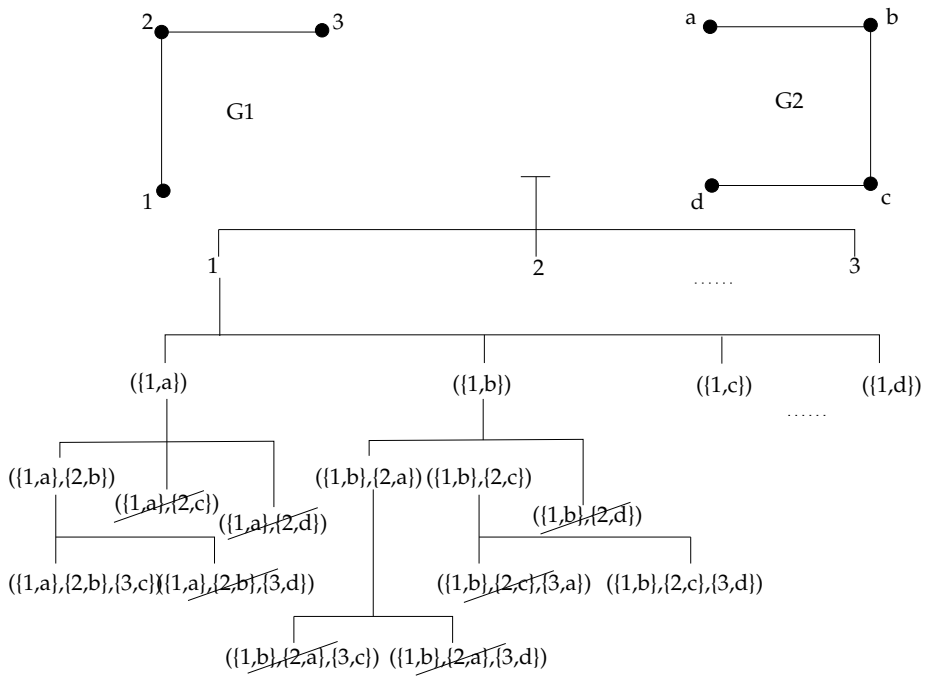


FIG. 2.5 – Arbre de recherche d'un sous-graphe commun maximum par énumération de tous les sous-graphes communs possibles. Les associations de sommets non valides sont barrées.

rer l'association sauvegardée et explorer une nouvelle extension. On continue, ainsi de suite, à explorer l'arbre en profondeur jusqu'à arriver au niveau des feuilles. Ces dernières, lorsqu'elles représentent des associations valides, correspondront à des sous-graphes communs maximaux, et la solution optimale sera une de ces feuilles dont le nombre de sommets est maximum. Il est à remarquer que chaque branche de l'arbre doit être explorée en entier car il est impossible de prévoir si une meilleure association de sommets existe ou non dans une branche non encore explorée.

2.3.2 Complexité théorique et performances pratiques

Conte et al. ont réalisé une étude expérimentale [12] visant à comparer les performances de trois algorithmes de recherche d'un sous-graphe commun induit maximum (MCIS). Les graphes testés ont été choisis parmi 6 catégories différentes et possèdent des ordres variant de 10 à 100 sommets. Le premier algorithme est une variante de l'algorithme de McGregor [50] adoptant l'approche par énumération de tous les sous-graphes communs. Les deux autres

Approche	Algorithme (variante)	Complexité en mémoire (au pire)	Complexité en temps (au pire)
Enumération des sous-graphes communs	McGregor[50]	$O(min(n_1, n_2))$	$\frac{(n_2+1)!}{(n_2-n_1+1)!}$
Réduction à la clique maximum	Durand et al.[20]	$O(n_1 \cdot n_2)$	$\frac{(n_2+1)!}{(n_2-n_1+1)!}$

TAB. 2.2 – Complexité au pire en mémoire et en temps de deux d'algorithmes de recherche de MCIS de deux graphes G_1 et G_2 d'ordres n_1 et n_2 respectivement. Table reproduite de [12].

algorithmes transforment le problème en un problème de recherche de clique maximum. La table 2.2 reproduit les complexités de deux de ces algorithmes.

Deux enseignements peuvent être tirés de cette étude des performances des deux principales approches :

- En théorie, les deux approches possèdent le même temps d'exécution au pire. En pratique, ces temps peuvent être accélérés grâce à des heuristiques et des techniques d'élagage qui permettent de n'explorer que les branches "prometteuses" de l'arbre de recherche [13].

- L'approche par énumération de tous les sous-graphes communs possibles est plus performante sur les graphes petits et/ou possèdant un grand alphabet d'étiquettes. Elle est aussi plus efficace lorsque les graphes sont peu denses. Dans tous les autres cas, le passage par la recherche de la clique maximum est plus efficace [12].

2.4 Conclusion

La solution exacte d'un grand nombre de problèmes posés dans des applications du monde réel correspond au calcul d'un sous-graphe commun maximum. Pourtant ce problème reste l'un des plus difficiles à vaincre du point de vue algorithmique théorique.

En pratique cependant, la nature et la taille des graphes manipulés font souvent que ce problème peut être résolu en des temps rapides. Il est donc crucial,

avant de choisir un algorithme ou une approche donnés, de bien s'imprégner des spécifités du domaine d'application et des propriétés des graphes comparés puisque ces informations sont précisément celles qui serviront à améliorer les performances et l'encombrement mémoire des algorithmes utilisés.

Nous avons vu au cours de ce chapitre que pour établir la similarité de deux sous-graphes d'ARN modélisant deux motifs tertiaires, nous avons besoin de calculer leur sous-graphe commun maximum. Ces sous-graphes d'ARN sont peu denses, ont moins de 40 sommets et un grand alphabet d'étiquettes. Il est donc plus efficace de choisir une approche par énumération exhaustive de tous les sous-graphes communs possibles. En outre, ce sous-graphe commun à deux sous-graphes d'ARN est particulier en ce sens qu'il doit maximiser le nombre d'arêtes de type non-canonique car il correspondrait alors à la structure consensus de la famille de motifs à laquelle appartiennent les deux sous-graphes similaires.

Il existe deux types de motifs tertiaires d'ARN :

- les *motifs locaux* incrustés dans des éléments de structure secondaire,
- les *motifs d'interaction* faisant interagir des régions distantes de la structure secondaire.

Bien que jouant des rôles structuraux parfois différents, ces deux types de motifs ont ceci en commun : ils sont composés essentiellement d'arêtes de type non-canonique et stabilisés occasionnellement par des arêtes de type empilement.

La similarité de deux sous-graphes d'ARN sera basée sur le calcul d'un sous-graphe commun ayant un nombre maximum d'arêtes de type non-canonique (dans le chapitre suivant, j'explique le pourquoi de cette définition). Pour tenir compte de cette condition, il s'est avéré plus approprié de mettre au point un algorithme *ad hoc* plutôt qu'adopter un algorithme standard de calcul d'un sous-grapphe commun maximum qui n'exploiterait pas de manière optimale les propriétés intrinsèques des graphes d'ARN.

Moyennant une adaptation légère, l'algorithme de calcul d'un sous-graphe commun non-canonique maximum que je propose s'applique avec succès aux deux types de motifs sus-cités et fournit, en pratique, une solution en un temps instantané. Cet algorithme n'est toutefois qu'un "module" dans une méthode globale d'extraction et de classification automatique de motifs tertiaires d'ARN.

Le chapitre suivant décrit cette méthode originale pour l'identification et la classification de *motifs locaux* d'ARN en commençant par la présentation de sa pierre angulaire : la mesure de similarité de deux sous-graphes d'ARN dont le calcul nécessite l'exécution d'un algorithme d'extraction d'un sous-graphe commun non-canonique maximum.

Chapitre 3

Extraction des motifs locaux

La méthode d'extraction automatique de motifs locaux récurrents faisant l'objet de ce chapitre est un pipeline qui prend en entrée une structure tertiaire d'ARN et renvoie en sortie un ensemble (pouvant être vide) de familles contenant chacune les occurrences similaires d'un motif local potentiel. Le terme "local" signifie "local à un élément de structure secondaire". Ce pipeline est composé de quatre modules (Figure 3.1).

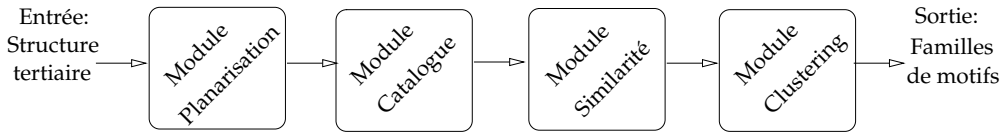


Fig. 3.1 – Vue d'ensemble de la méthode d'extraction et de classification de motifs locaux récurrents illustrée par un agencement de quatre modules. La sortie de chaque module représente l'entrée du module suivant.

La structure tertiaire, qui est un fichier au format PDB, est passée à un programme d'annotation pour produire son graphe d'ARN. De ce graphe, on élimine les pseudo-noeuds [1] et on supprime temporairement les arêtes de type non-canonique (Module **Planarisation**). Ce qui reste du graphe d'ARN est donc le graphe de la structure secondaire sans pseudo-noeuds. Ce dernier peut, à son tour, être modélisé par un arbre duquel on va extraire les éléments de structure secondaire qui sont les renflements, les boucles internes, les jonctions et les boucles terminales. A chacun de ces éléments de structure secondaire, on restore l'ensemble de ses arêtes non-canoniques. On obtient ainsi une liste ou *catalogue* d'éléments de structure secondaire don-

[1] Un pseudo-noeud se forme lorsque deux régions en simple brin n'appartenant pas à la même hélice forment des liaisons de type Watson-Crick.

nés sous forme de sous-graphes d'ARN (Module **Catalogue**). Pour chaque paire d'éléments dans le catalogue, on calcule une mesure de similarité basée sur le calcul d'un *sous-graphe non-canonique commun maximum* (Module **Similarité**). A partir de toutes les mesures de similarités calculées deux à deux, on remplit une matrice de dissimilaritées sur laquelle on lance une classification hiérarchique (Module **Clustering**). On obtient ainsi un arbre de classification sur lequel on va identifier les sous-arbres contenant les éléments fortement similaires d'après un seuil de similarité fixé. Chacun de ces sous-arbres ou "clusters" représente une famille potentielle de motifs locaux. A la fin, on considère ces sous-arbres un à un, et pour chacun d'eux, on extrait le sous-graphe non-canonique commun aux membres du cluster et on le renvoie comme "noyau" non-canonique de la structure consensus de la famille trouvée. L'ensemble des séquences de tous les membres du cluster correspondra à la séquence signature de cette famille.

La qualité de cette classification dépend entièrement de la notion de similarité de deux sous-graphes d'ARN. Celle-ci est basée sur le calcul d'un sous-graphe commun ayant une propriété particulière : il possède un nombre maximum d'arêtes non-canoniques. Cette idée fut suggérée par l'observation suivante : la structure non-canonique commune à deux sous-graphes d'ARN représente le "noyau" non-canonique de la structure consensus d'un motif potentiel. Dans cette structure consensus, les arêtes sont préférablement empilées, c'est à dire reliées par un lien squelette sucre-phosphate (cf. définition de motif tertiaire, section 1.1). Si, de plus, ce noyau se trouve contigu à une arête canonique flanquant le motif, cette arête sera ajoutée à la structure consensus (Figure 3.2).

Cette structure non-canonique commune à deux sous-graphes d'ARN sera appelée, par la suite, *"Largest Extensible Common Non-canonical Subgraph (LECNS)"*. Elle est la base de la mesure de similarité de deux motifs locaux.

Dans la section 1, je définis la mesure de similarité de deux sous-graphes d'ARN qui est au coeur du module **Similarité**. La section 2 reprend en détail la description de la méthode d'extraction des motifs locaux. La section 3 décrit les résultats obtenus. Enfin, la section 4 présente la suite de programmes implémentant cette méthode.

3.1 Similarité de deux motifs locaux

La similarité de deux sous-graphes d'ARN modélisant deux motifs locaux potentiels nécessite le calcul de leur *plus grand sous-graphe non-canonique*

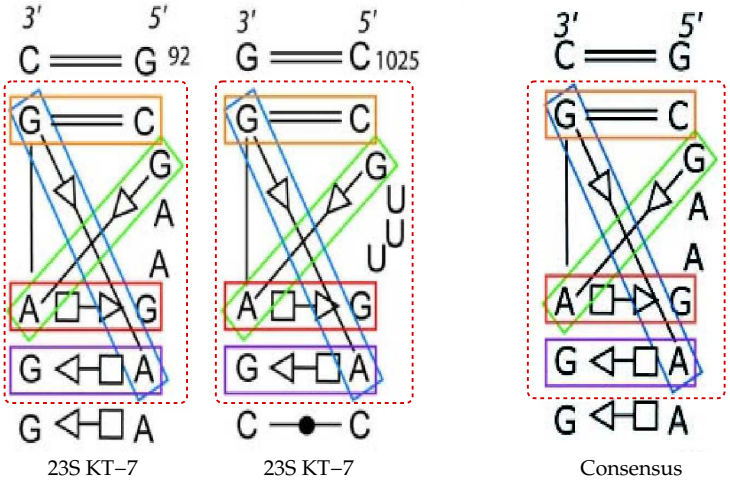


FIG. 3.2 – Les arêtes non-canoniques communes à deux ou plusieurs occurrences d'un motif (ici le Kink-turn) forment le noyau non-canonique de sa structure consensus (encadrée en rouge). Figure extraite de [45] avec adaptation.

commun extensible ("Largest Extensible Common Non-canonical Subgraph (LECNS)").

3.1.1 Définitions

Définition 2 La **signature non-canonique** d'un sous-graphe d'ARN est la chaîne de caractères obtenue par concaténation des étiquettes de ses arêtes de type non-canonique ordonnées par ordre alphabétique.

Définition 3 La **taille non-canonique** d'un sous-graphe d'ARN G, notée $||G||$, est le nombre de ses arêtes de type non-canonique.

Définition 4 Un **sous-graphe non-canonique** d'un sous-graphe d'ARN est un sous-graphe dont toutes les arêtes sont de type non-canonique.

Définition 5 Soit g un sous-graphe non-canonique d'un sous-graphe d'ARN G. La **complétude** de g dans G est le graphe obtenu en ajoutant à g toutes les arêtes de G de type canonique et squelette sucre-phosphate ("backbone") ayant au moins une extrémité dans g.

Définition 6 Un sous-graphe non-canonique G_{12} commun à deux sous-graphes d'ARN G_1 et G_2 est **extensible** si ses complétudes dans G_1 et G_2, respectivement, sont isomorphes.

Définition 7 Un **LECNS** ("Largest Extensible Common Non-canonical Subgraph") de deux sous-graphes d'ARN G_1 et G_2, noté LECNS(G_1,G_2), est un sous-graphe non-canonique commun extensible de taille (non-canonique) maximum.

Note : Un LECNS de deux sous-graphes d'ARN n'est pas nécessairement connexe.

La figure 3.3 illustre toutes ces définitions.

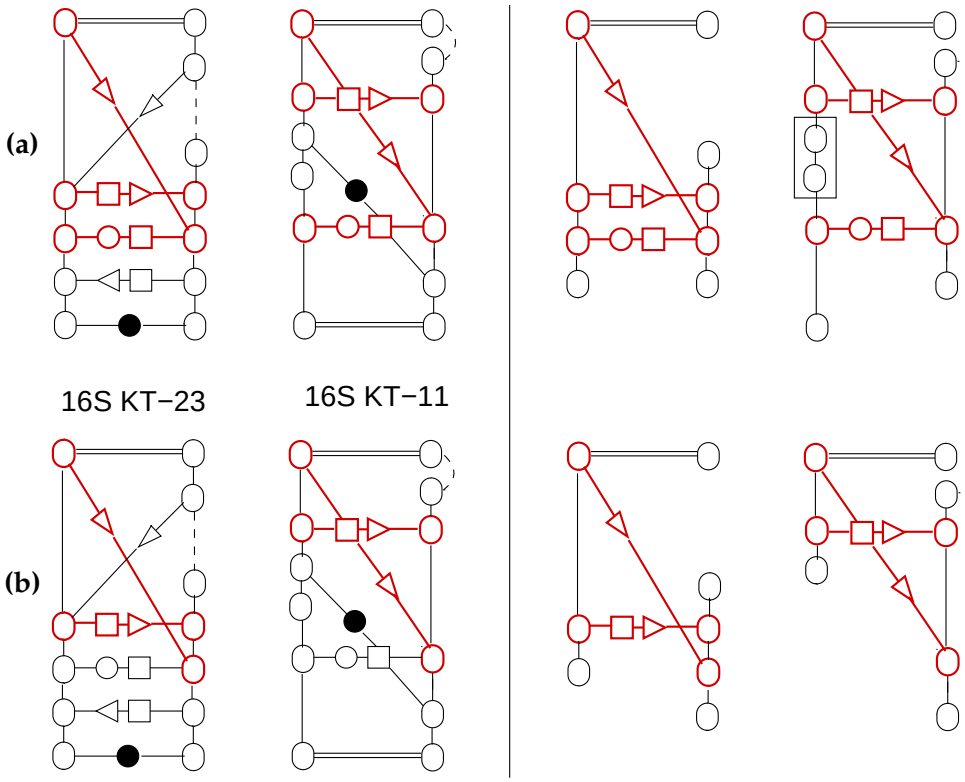


FIG. 3.3 – Deux sous-graphes d'ARN correspondant à deux éléments de structure secondaire contenant des motifs Kink-turn de l'ARNr 16S tels que illustrés dans [45]. Les traits en pointillés indiquent des nucléotides en simple brin. Dans (a) à gauche, un sous-graphe non-canonique commun de taille 3 apparait en rouge. Les complétudes de ce sous-graphe non-canonique commun montrées à droite ne sont pas isomorphes. Dans (b), un autre sous-graphe non-canonique commun de taille 2 apparait en rouge (à gauche) dont les complétudes (à droite) sont isomorphes. Donc un LECNS de ces deux sous-graphes d'ARN est de taille (non-canonique) 2.

3.1.2 Mesure de similarité

La similarité de deux sous-graphes d'ARN G_1 et G_2, notée $sim(G_1, G_2)$, est définie par :

$$sim(G_1, G_2) = \begin{cases} \dfrac{||LECNS(G_1, G_2)||}{max(||G_1||, ||G_2||)} & \text{si } ||LECNS(G_1, G_2)|| > 1 \\ 0 & \text{sinon.} \end{cases} \quad (3.1)$$

Un LECNS formé d'une seule arête non-canonique n'est pas considéré comme un noyau de motif valide, d'où la condition $||LECNS(G_1, G_2)|| > 1$.

La mesure de similarité sim vérifie les propriétés suivantes :

- $0 \leq sim(G_1, G_2) \leq 1$,
- $sim(G_1, G_2) = sim(G_2, G_1)$ (Symétrie)
- $sim(G_1, G_2) = 1 \Longrightarrow$ les complétudes des plus grands sous-graphes non-canoniques de G_1 et G_2 sont isomorphes.
- $sim(G_1, G_2) = 0 \Longrightarrow G_1$ et G_2 n'ont pas de sous-graphe non-canonique commun extensible de taille > 1.

De manière duale, la mesure de dissimilarité de deux sous-graphes d'ARN, notée $dis(G_1, G_2)$, est définie par :

$$dis(G_1, G_2) = 1 - sim(G_1, G_2)$$

3.1.3 Implémentation

Le graphe d'ARN qui modélise une structure tertiaire est représenté en mémoire par une liste d'adjacence. Dans ce graphe, le nombre d'arêtes incidentes à un sommet est borné par la constante 7. En effet, de chaque nucléotide (i.e. un sommet dans le graphe d'ARN) peuvent être issus :

- au plus deux liens backbone (orientés $5'$ et $3'$ respectivement),
- au plus trois interactions impliquant les trois côtés de la base : W, H et S.
- et au plus deux interactions de type empilement 55 et 33 (Note : l'empilement 35 ou 53, spécifique au squelette sucre-phosphate, n'a pas été considéré car il ne constitue pas un type d'interaction remarquable).

Un sous-graphe d'ARN, lui, est représenté sous forme d'un enregistrement où sont stockés :

- l'ordre du sous-graphe,
- la taille du sous-graphe,
- l'ensemble des identifiants des sommets donné par une liste chaînée, ordonnée arbitrairement,
- l'ensemble des identifiants des arêtes donné par une liste chaînée, ordonnée arbitrairement,
- la signature non-canonique du sous-graphe.

Dans le choix de cette dernière structure de données, l'optimisation de l'espace mémoire a été priviligiée par rapport au temps de parcours des listes.

I) Procédure Complétude

Soient un sous-graphe non-canonique $g = (W, F)$ d'un sous-graphe d'ARN $G = (V, E)$. La procédure *Complétude*(g, G) retourne la complétude de g dans G.

Procédure Complétude
Entrée : Sous-graphes $g = (W, F)$ et $G = (V, E)$
Sortie : Sous-graphe d'ARN $g' = (W', F')$

---

1 Début
2 $W' \leftarrow W$, $F' \leftarrow F$
3 Pour chaque sommet w dans W faire
4 Pour chaque sommet v voisin de w dans V faire
5 Si (l'arête (w, v) a une étiquette de type canonique ou "backbone"
6 et v n'est pas dans W) alors
7 $W' \leftarrow W' \cup \{v\}$;
8 $F' \leftarrow F' \cup \{(w, v)\}$;
9 Fin pour ;
10 Fin pour ;
11 Retourner $g' = (W', F')$
12 Fin

Complexité au pire :

- *En temps* :

Le parcours des sommets de W se fait en $\mid W \mid$ étapes. Le parcours des sommets v de V voisins de w se fait en $O(1)$. Le test du type d'une étiquette

d'arête se fait en $O(1)$ et le test de l'appartenance d'un sommet à W doit parcourir tout W. Ainsi, la complexité en temps au pire de la procédure *Complétude*(g, G) est en $O(\mid W \mid^2)$.

- *En espace* :
L'espace mémoire nécessaire pour représenter un sous-graphe d'ARN $G = (V, E)$ est en $O(\mid V \mid)$. La complexité en mémoire de la procédure *Complétude*(g, G) est donc en $O(\mid W \mid + \mid V \mid)$ (i.e. espace requis pour stocker les sous-graphes g et G).

II) Procédure LECNS

Le principe de fonctionnement de cette procédure est le suivant : on compare en parallèle les combinaisons d'arêtes non-canoniques appartenant aux deux sous-graphes. On ne s'intéresse qu'aux paires de combinaisons qui ont des ensembles identiques d'étiquettes. Si ces deux combinaisons d'arêtes possèdent des complétudes isomorphes, alors un sous-graphe non-canonique commun extensible est trouvé. Comme il s'agit d'obtenir un sous-graphe extensible de taille maximum, on commence par explorer les combinaisons d'arêtes non-canoniques dans l'ordre décroissant de leurs tailles. La recherche s'arrête au premier LECNS trouvé.

a) Notations

Soient G_1 et G_2 deux sous-graphes d'ARN. Notons :
- n_1 (resp. n_2) l'ordre de G_1 (resp. G_2),
- m_1 (resp. m_2) la taille de G_1 (resp. G_2),
- $G_1.ncsig$ (resp. $G_2.ncsig$) la signature non-canonique de G_1 (resp. G_2),
- S_{12} l'ensemble des caractères obtenu par concaténation sans redondance des étiquettes communes à $G_1.ncsig$ et $G_2.ncsig$, ordonné par ordre alphabétique,
- E_1 (resp. E_2) le tableau contenant les identifiants, ordonnés arbitrairement, des arêtes de G_1 (resp. G_2) dont les étiquettes sont dans S_{12},
- *Générer_partie*(E_1, i) (resp. *Générer_partie*(E_2, i)) une fonction qui génère exhaustivement et sans redondance les parties de E_1 (resp. E_2) de taille i. A chaque appel, elle retourne un pointeur sur une nouvelle partie de E_1 (resp. E_2) de taille i s'il reste des parties à générer. Retourne NULL sinon,
- p_1 (resp. p_2) un pointeur sur une partie de E_1 (resp. E_2),
- $p_1.graphe$ (resp. $p_2.graphe$) le graphe induit par les arêtes d'une partie de E_1 (resp. E_2) pointée par p_1 (resp. p_2).
- *Mapping* un ensemble (initialement vide) de deux listes de tuples d'identi-

fiants de sommets et d'arêtes retournés par la procédure *Isomorphisme*.

b) Pseudo Code

Procédure LECNS

Entrée : Sous-graphes G_1 et G_2, Ensemble de caractères S_{12}
Sortie : $Mapping$ initialisé à $\emptyset$

---

1 Début
2 $E_1 \leftarrow$ identifiants des arêtes de G_1 dont les étiquettes sont dans S_{12}
3 $E_2 \leftarrow$ identifiants des arêtes de G_2 dont les étiquettes sont dans S_{12}
4 $Taillemin \leftarrow min(\mid E_1 \mid, \mid E_2 \mid)$
5 Pour $i = Taillemin$ à 2 faire
6 $p_1 \leftarrow$ *Générer_partie*(E_1, i) ;
7 Tant que $(p_1 \neq NULL)$ faire
8 $g_1 \leftarrow p_1.graphe$;
9 $p_2 \leftarrow$ *Générer_partie*(E_2, i) ;
10 Tant que $(p_2 \neq NULL)$ faire
11 $g_2 \leftarrow p_2.graphe$;
12 Si $(g_1.ncsig = g_2.ncsig)$ alors
13 $g_1 \leftarrow$ *Complétude* (g_1, G_1) ;
14 $g_2 \leftarrow$ *Complétude* (g_2, G_2) ;
15 Si $(g_1.ordre = g_2.ordre$ et $g_1.taille = g_2.taille)$ alors
16 $Mapping \leftarrow Isomorphisme(g_1, g_2)$;
17 Si $Mapping \neq \emptyset$ alors retourner $Mapping$
18 sinon $p_2 \leftarrow$ *Générer_partie*(E_2, i) ;
19 sinon $p_2 \leftarrow$ *Générer_partie*(E_2, i) ;
20 sinon $p_2 \leftarrow$ *Générer_partie*(E_2, i) ;
21 Fin tant que ;
22 $p_1 \leftarrow$ *Générer_partie*(E_1, i) ;
23 Fin tant que ;
24 Fin pour ;
25 Retourner $Mapping$;
26 Fin.

---

c) Explication de l'algorithme

Lignes 2 et 3

L'algorithme commence par réduire l'ensemble des arêtes à examiner dans

chaque sous-graphe d'entrée aux ensembles E_1 et E_2 d'arêtes dont les étiquettes sont dans S_{12}. En effet, si G_1 et G_2 ont un sous-graphe non-canonique commun, alors sa signature non-canonique est forcément incluse dans S_{12}. De plus, sa taille sera bornée par le cardinal $Taillemin$ du plus petit des ensembles E_1 et E_2.

Lignes 6 à 23

On commence par générer une première partie de E_1 de taille i et une première partie de E_2 de même taille i, stockées respectivement dans p_1 et p_2. Puis on teste si les signatures non-canoniques des graphes $p_1.graphe$ et $p_2.graphe$ sont égales. Si oui, on calcule les complétudes de $p_1.graphe$ dans G_1 et $p_2.graphe$ dans G_2 respectivement puis on teste leur isomorphisme par appel de la procédure *Isomorphisme*. Si les complétudes sont isomorphes, leur $Mapping$ est retourné et l'algorithme s'arrête. Sinon, on génère une nouvelle partie de E_2 pour la comparer avec la première partie de E_1 (exécuter les lignes 11 à 15). Si cette comparaison trouve un $Mapping$ non vide, on le retourne et l'algorithme s'arrête. Sinon, on génère une nouvelle partie de E_2 et la compare à nouveau avec la première partie de E_1.

Lorsque, pour une partie donnée de E_1 de taille i, on épuise toutes les parties de E_2 de taille i sans trouver de $Mapping$ non vide, on génère une nouvelle partie de E_1 de taille i que l'on va comparer successivement avec les parties de E_2 de même taille i dans l'ordre de leur génération. L'algorithme procède ainsi jusqu'à trouver un $Mapping$ non vide (le retourner alors et s'arrêter), ou bien retourner un $Mapping$ vide et s'arrêter après avoir effectué toutes les comparaisons de paires de parties de E_1 et E_2 de toutes les tailles allant de $Taillemin$ à 2.

Pour s'assurer que le premier $Mapping$ trouvé corresponde à un sous-graphe non-canonique commun de *taille maximum*, on génère les parties de E_1 et E_2 dans l'ordre décroissant de leurs tailles, en commençant par $Taillemin$ et descendant jusqu'à 2.

L'algorithme LECNS retourne le *premier* sous-graphe non-canonique commun extensible maximum trouvé, ce qui s'est avéré suffisant en pratique pour détecter une similarité entre deux occurrences d'un même motif. Cependant, il est aisé de le modifier de manière à renvoyer *tous* les sous-graphes communs extensibles de (même) taille maximum, notée $taillemax$. Pour cela, il suffit de générer et tester toutes les parties de E_2 de taille $taillemax$.

La fonction *Générer_partie*(E_k, i) $(k = 1, 2)$ peut être implémentée en utilisant un au choix des algorithmes d'énumération des combinaisons de i parmi n $(i < n)$ vérifiant les propriétés de changement minimal et sans boucle ("Minimal Change and Loopless Properties"). Une génération de combinaisons vérifie la propriété de changement minimal si toute combinaison, à l'exception de la première, est générée à partir de la précédente en remplaçant un élément seulement. Un algorithme de génération de combinaisons est dit sans boucle si le nombre maximum d'opérations nécessaires pour générer une combinaison est une constante indépendante de n et k. Deux algorithmes d'énumération de combinaisons vérifiant ces deux propriétés sont décrits dans [35].

L'algorithme d'*Isomorphisme* utilisé est celui de Valiente [67]. Dans sa version originale, cet algorithme produit *tous* les *Mappings* possibles. Cependant, l'algorithme LECNS utilise une version modifiée de l'algorithme de Valiente dans laquelle *un seul Mapping* est retourné (le premier trouvé).

Le problème d'isomorphisme de graphes n'est ni NP-complet ni polynomial [67]. Toutefois plusieurs algorithmes ont été proposés pour calculer l'isomorphisme de deux graphes. Cinq parmi les plus connus d'entre eux ont été implémentés dans la bibliothèque LEDA [51]. Dans un comparatif de leurs performances [62] sur une base de données de graphes particuliers [2] [21] l'algorithme de Valiente n'apparait pas comme le plus rapide. Cependant, dans le contexte de nos sous-graphes d'ARN dont l'ordre ne dépasse pas 40 sommets et l'alphabet d'étiquettes est de cardinal 14, cet algorithme s'est avéré suffisamment rapide.

Afin d'identifier la séquence signature du motif, seules les étiquettes des arêtes sont prises en considération dans la recherhe d'un isomorphisme. Ainsi, si deux arêtes dans G_1 et G_2 ont même orientation et même étiquette, elles seront associées même si les sommets aux extrémités ne possèdent pas les mêmes étiquettes.

[2] Graphes connectés aléatoirement, graphes filets 2D, 3D, 4D et graphes à degré borné.

d) Complexité au pire

- *En temps* :

Supposons, sans perte de généralité, que $n_1 \leq n_2$. L'algorithme d'isomorphisme de Valiente a une complexité en temps au pire de $O((n_1+1)!)$ [67]. Je rappelle que cette complexité est celle de la version originale qui renvoie tous les mappings possibles. L'algorithme LECNS se contente, quant à lui, du premier mapping trouvé. En pratique, ceci réduit significativement le temps de calcul.

L'exécution des lignes 2 et 3 de la procédure LECNS nécessite de parcourir toutes les arêtes de G_1 et G_2 pour construire les ensembles E_1 et E_2. Elle a donc une complexité en $O(m_1+m_2)$.

La boucle d'exécution des comparaisons appariées (à partir de la ligne 5) effectue au pire $(Taillemin \cdot \sum_{i=2}^{Taillemin} \binom{|E_1|}{i} \cdot \sum_{i=2}^{Taillemin} \binom{|E_2|}{i})$ fois deux appels de la procédure *Complétude* et un appel de la procédure *Isomorphisme*. Ces deux procédures travaillent sur des sous-graphes de G_1 et G_2. Les ordres et tailles de ces sous-graphes sont bornés, respectivement, par les ordres et tailles de G_1 et G_2. Cette dernière étape coûte donc au pire :

$$O(Taillemin \cdot \sum_{i=2}^{Taillemin} \binom{|E_1|}{i} \cdot \sum_{i=2}^{Taillemin} \binom{|E_2|}{i} \cdot (n_1^2+n_2^2+(n_1+1)!).$$

Le coût total de la procédure LECNS est donc en :

$$O\left(Taillemin \cdot \sum_{i=2}^{Taillemin} \binom{\mid E_1 \mid}{i} \cdot \sum_{i=2}^{Taillemin} \binom{\mid E_2 \mid}{i} \cdot \left(n_1^2+n_2^2+(n_1+1)!\right)\right)$$

- *En espace* :

La procédure LECNS compare deux sous-graphes G_1 et G_2 nécessitant un espace en $O(n_1+n_2)$.

Les lignes 2 et 3 stockent les identifiants des arêtes de E_1 et E_2. L'espace requis est donc en $O(\mid E_1 \mid + \mid E_2 \mid)$.

A partir de la ligne 5, le pire scénario pour chaque itération se produit lorsque les deux parties générées puis comparées ont la même signature non-canonique et leurs complétudes possèdent même ordre et même taille. Dans ce cas, la procédure *Isomorphisme* sera appelée. Cette dernière nécessite un

espace en $O(n_1^2)$ [67].

La fonction *Générer_partie*(E_k, i) $(k = 1, 2)$ peut être implémentée en utilisant l'algorithme à pile non-récursif de [35]. Dans ce cas, elle nécessitera une pile de taille i, i étant borné par $Taillemin$.

La complexité au pire en espace de la procédure LECNS est donc en $O(n_1^2)$.

e) En pratique

L'algorithme LECNS (implémenté par une version moins optimisée en encombrement mémoire) a été exécuté sur des graphes d'ARN de structures ribosomiques dont les trois plus grandes sont :
- l'ARNr 16S de *T.thermophilus* (code PDB 1J5E) : 1513 nucléotides,
- l'ARNr 23S de *H.marismortui* (code PDB 1S72, chaîne 0) : 2754 nucléotides,
- l'ARNr 23S de *E.coli* (code PDB 2AW4, chaîne B) : 2841 nucléotides.

Le temps d'exécution le plus long a été enregistré pour deux sous-graphes (représentant deux jonctions de l'ARNr 23S de *E.coli*) ayant respectivement 27 et 36 sommets. Ce temps est de 0.48 secondes obtenu sur un Pentium 4 de fréquence 2.4GHZ.

3.2 Méthode d'extraction des motifs

3.2.1 Module Planarisation

Des structures cristallographiques d'ARN ribosomique ont été téléchargées à partir de la Nucleic acid DataBase (NDB) [6], une base de données dédiée aux acides nucléiques.

Ces structures sont fournies en entrée au programme d'annotation *Rnaview* [74] qui produit le graphe d'ARN correspondant à la structure entrée. Autrement dit, toutes les interactions présentes dans une structure tertiaire annotées selon la nomenclature Leontis-Westhof.

De ce graphe, on supprime les pseudo-noeuds en utilisant le programme *Secrna* développé par Y. Ponty [59] et on supprime provisoirement les arêtes de type non-canonique. Cette étape a pour but de réduire le graphe d'ARN au graphe de la structure secondaire sans pseudo-noeuds.

3.2.2 Module Catalogue

Ce graphe de la structure secondaire sans pseudo-noeuds peut, à son tour, être modélisé par une structure d'arbre [28, 1] comme illustré par la figure 3.4. Dans cet arbre, une base libre est représentée par une feuille et une paire de bases par un noeud interne.

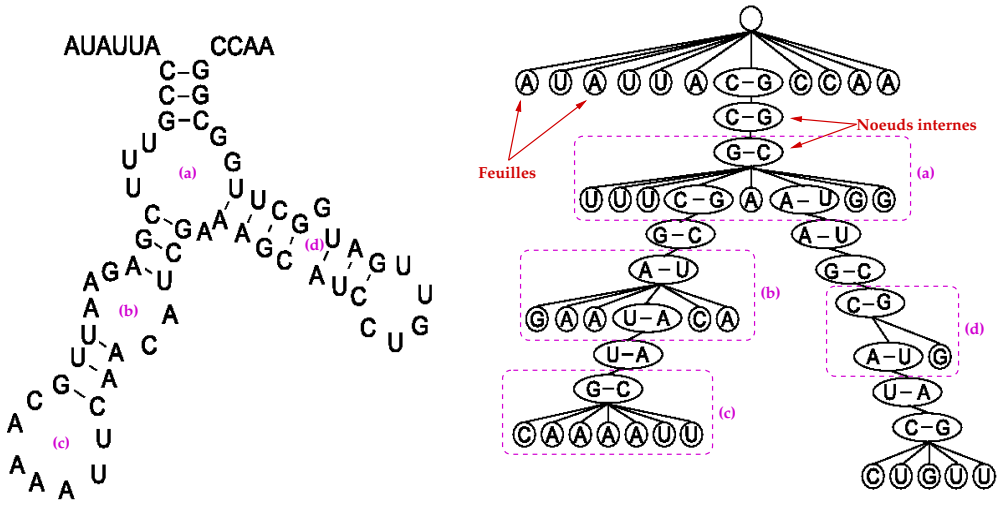


FIG. 3.4 – Arbre représentant une structure secondaire d'ARN.(a) triple jonction. (b) boucle interne. (c) boucle terminale. (d) renflememt droit. Figure extraite de [1] avec adaptation.

Une fois cet arbre construit, on extrait les noeuds internes qui correspondent aux renflements, boucles internes, jonctions et boucles terminales.

Une boucle terminale est un noeud interne dont tous les fils sont des feuilles. Un renflement droit (resp. gauche) est un noeud interne dont le dernier fils (resp. le premier fils) est un noeud interne et tous les autres fils des feuilles. Une jonction est un noeud interne ayant au moins deux fils noeuds internes et un nombre quelconque (pouvant être nul) de fils feuilles. Une boucle interne est un noeud interne ayant exactement un fils noeud interne et un premier et un dernier fils feuilles.

L'algorithme utilisé pour la construction de l'arbre est ceui de J. Allali [1] :

Procédure Arbre

Entrée : Graphe de la structure secondaire d'ordre N

Sortie : Racine r de l'arbre

Début
% Remplir une table T de même taille N que la séquence (i.e. ordre du graphe),
% ordonnée par ordre croissant de l'extrémité $5'$ vers l'extremité $3'$.
Pour $i = 1$ à N faire
  $T[i] \leftarrow j$ si i est apparié à j
  $T[i] \leftarrow -1$ si i est non apparié
Fin pour ;
% Parcourir T et construire l'arbre sous forme d'une pile :
Créer une racine r et l'empiler.
Pour $i = 1$ à N faire
  si $(T[i] = -1)$ créer une feuille et l'ajouter comme fils
        du noeud interne au sommet de la pile ;
  si $(T[i] > i)$ créer un noeud interne et l'ajouter comme fils
        du noeud interne au sommet de la pile ;
  si $(T[i] < i)$ dépiler le sommet de la pile ;
Fin pour ;
Retourner le sommet de la pile r ;
Fin.

Maintenant à chacun de ces noeuds internes (représentant des éléments de structure secondaire), on restitue toutes les arêtes de type non-canonique reliant deux quelconques de ses nucléotides. On obtient ainsi un *catalogue* ou liste d'éléments de structure secondaire enrichis de toute l'information sur leurs interactions tertiaires locales (Figure 3.5).

Des exemples de catalogues sont disponibles à l'adresse :

`http://www.lri.fr/~md/RNA/CATALOGUE/catalogue.htm`

Note : On remarquera que les arêtes de type empilement n'apparaissent pas dans ces catalogues. La raison est que l'annotation de *Rnaview* n'adopte pas la classification décrite dans la section 1.2.2. Cette dernière a été proposée dans l'annotateur FR3D [61] à une date postérieure à la réalisation de ce travail (Juillet 2007).

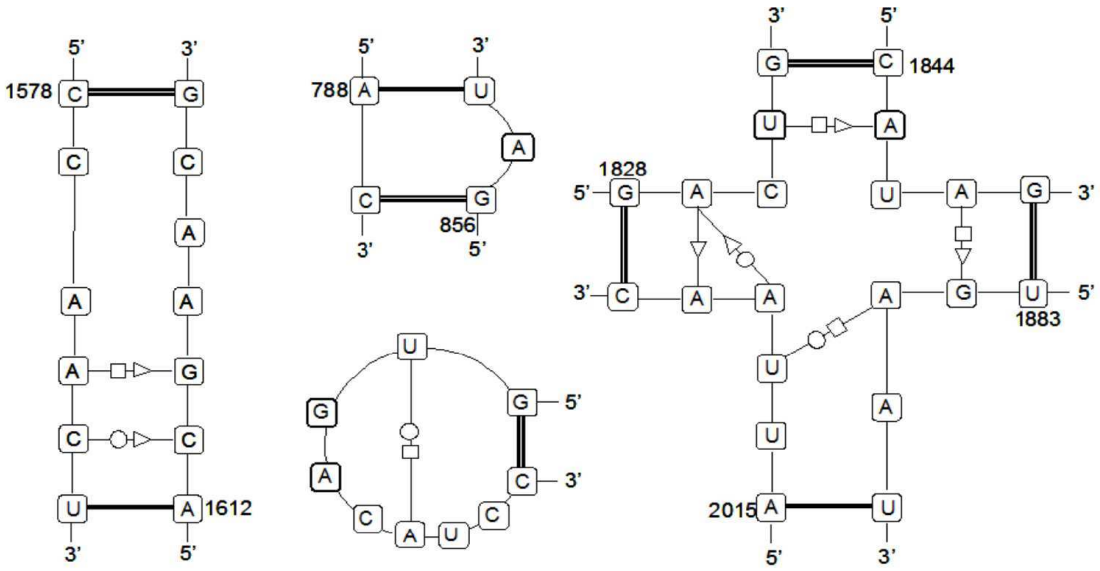


FIG. 3.5 – Exemples d'éléments de structure secondaire tels que produits par le module Catalogue.

3.2.3 Module Similarité

Pour chaque paire d'éléments du catalogue, on teste s'ils ont en commun au moins deux étiquettes non-canoniques. Cette condition exprime le fait qu'une structure consensus, si elle existe, doit être composée d'au moins deux arêtes non-canoniques, puisqu'une seule arête non-canonique n'est pas suffisante pour former un motif biologiquement valide.

Si les deux éléments du catalogue comparés vérifient cette condition, la procédure LECNS est appelée et une mesure de dissimilarité est calculée en utilisant la formule définie dans la section 3.1.2. On remplit ainsi une matrice de dissimilarités qui sera envoyée en entrée au module suivant.

3.2.4 Module Clustering

Les méthodes de clustering utilisent la dissimilarité entre objets pour former des clusters d'objets similaires. Ces méthodes n'exigent pas que cette dissimilarité soit une distance (i.e. métrique). Le choix de cette mesure dépend de la nature des données et du pouvoir de cette dernière à "capturer" l'information la plus pertinente dans l'évaluation de la similarité.

Dans ce module, on applique sur la matrice de dissimilarités un **clustering hiérarchique**. Le choix de ce type de clustering sera mieux compris après avoir présenté son principe général :

Initialement, chaque individu (i.e. sous-graphe d'ARN) forme un cluster. On cherche à réduire le nombre de clusters en procédant itérativement. A chaque étape, on fusionne les clusters les plus similaires en utilisant une règle de lien (dite aussi d'agglomération) basée sur une mesure de distance entre clusters. Notons que cette mesure de distance ne doit pas être confondue avec la métrique définie dans la section 2.2.2.

Lorsque les clusters sont formés d'objets individuels, la distance entre clusters est égale à la mesure de dissimilarité choisie. Si les clusters sont formés de plusieurs individus, diverses possibilités existent. Par exemple, la distance entre deux clusters sera égale à la distance entre les deux objets (appartenant chacun à un cluster) qui sont les plus proches voisins ("nearest neighbors"). Une autre mesure de lien consiste à calculer la distance moyenne entre toutes les paires d'individus dans les deux clusters. Cette méthode est connue sous le nom de UPGMA ("Unweighted Pair-Group Methods using arithmetic Averages") [63].

Le résultat d'un clustering hiérarchique est une représentation graphique sous forme d'arbre appelée *dendrogramme* (Figure 3.7). Les feuilles de l'arbre sont les individus alignés sur l'axe des abcisses. Les distances de lien sont indiquées en ordonnées. Lorsque deux clusters sont liés par une distance d, deux traits verticaux sont tracés des abcisses des clusters jusqu'à l'ordonnée d puis ils sont reliés par un trait horizontal.

Ainsi, *toutes* les relations de proximité entre individus apparaissent clairement sur l'arbre, permettant d'y repérer les éventuelles incohérences de classification. C'est cet avantage, non offert par d'autres types de clustering (ex. le clustering par partitionnement), qui a motivé notre choix du clustering hiérarchique pour classifier nos motifs d'ARN. Quant à la règle d'agglomération, la méthode UPGMA a été choisie car, de toutes les règles de lien testées, elle a été celle qui a permis d'obtenir des clusters bien différenciés pour les motifs connus.

Une fois le dendrogramme produit, l'étape suivante consiste à couper l'arbre au niveau d'un seuil de dissimilarité de manière à extraire des clusters regroupant distinctement des motifs biologiquement valides.

Cette dissimilarité-seuil a été fixée de manière empirique en observant les occurrences de motifs connus, répertoriées dans [40, 45]. Les familles (distinctes) de motifs présentant des similarités sont la boucle E et la boucle Sarcin-ricin [40]. Un seuil de dissimilarité de 0.4 (i.e. une similarité de 0.6) s'est avéré le plus optimal pour éviter de regrouper une variante de la boucle E avec une variante perturbée de la boucle Sarcin-ricin (Figure 3.6).

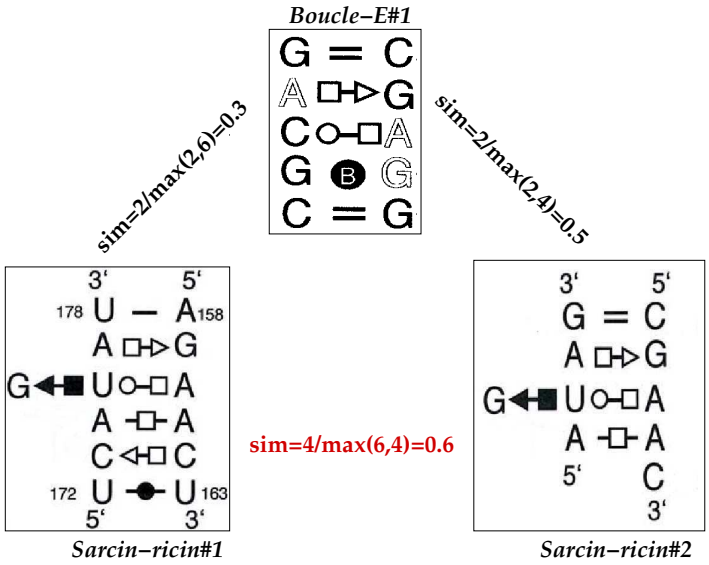


FIG. 3.6 – Seuil de similarité optimal pour regrouper des occurrences d'une même famille, ici le motif Sarcin-ricin.

La figure 3.7 montre le dendrogramme d'un clustering hiérarchique utilisant la méthode UPGMA appliqué au catalogue de la chaîne 23S de l'ARN ribosomique de l'organisme *H. marismortui*. Ce dendrogramme a été produit en utilisant la fonction *hclust* du paquetage *R project for statistical computing* [3]. Les clusters, obtenus à partir d'un seuil de dissimilarité de 0.4, représentent des motifs potentiels.

[3]http ://www.r-project.org/

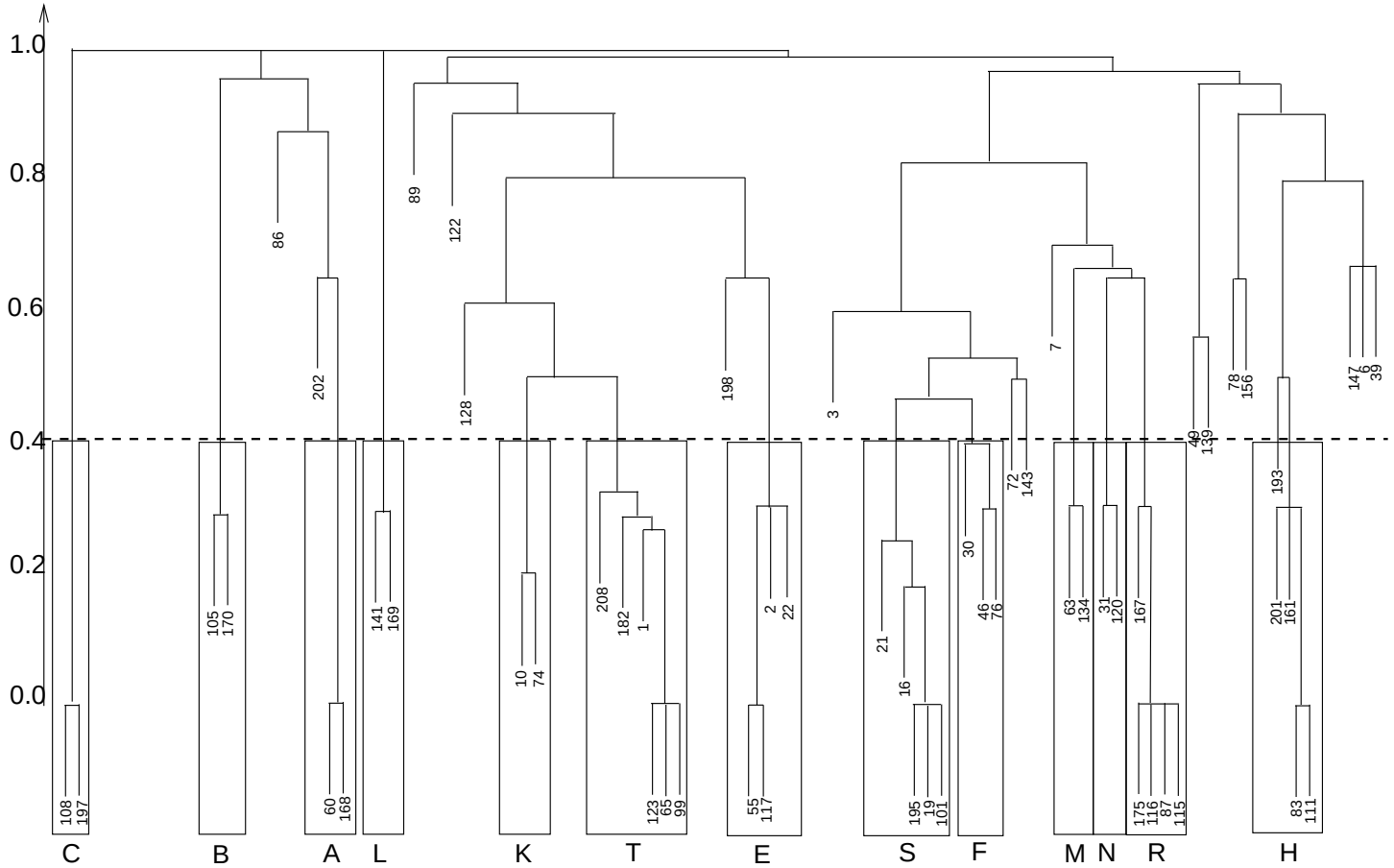


FIG. 3.7 – Dendrogramme de la chaîne 23S de l'ARN ribosomique de *H. marismortui* (code PDB 1S72). Les clusters obtenus à partir d'un seuil de dissimilarité 0.4 sont encadrés et étiquetés par des lettres de l'alphabet. Lorsqu'un cluster correspond à un motif structural connu, cette lettre est la première de son nom. Exemple "S" pour le motif "Sarcin-ricin", "K" pour le motif "Kink-turn", etc.

3.3 Résultats

Afin de vérifier que la méthode redécouvre automatiquement et sans *a priori* les motifs répertoriés dans [40, 45], les mêmes structures ribosomiques cristallographiques ont été utilisées :

- l'ARNr 23S et 5S de *H. marismortui* (code PDB 1S72),
- l'ARNr 23S et 5S de *E.coli* (code PDB 2AW4),
- et l'ARNr 16S de *T. thermophilus* (code PDB 1J5E).

Les résultats du clustering ont révélé que la mesure de similarité choisie représente un bon indicateur de la similarité de deux occurrences d'un même motif pourvu que le nombre d'arêtes non-canoniques qu'elles ont en commun contribue d'au moins 2/3 dans la taille non-canonique du plus grand de leurs sous-graphes correspondants. Dans le cas contraire, la valeur du dénominateur dans la formule de similarité fera baisser la valeur de celle-ci. Deux occurrences comportant une structure commune identique seront ainsi considérées comme dissimilaires.

Pour traiter ces cas particuliers, une étape supplémentaire a été ajoutée : une fois le dendrogarmme construit et les clusters formés avec le seuil optimal, on identifie pour chaque cluster un sous-graphe commun non-canonique appelé *noyau non-canonique* et on le compare avec les sous-graphes non clusterisés avec le seuil optimal. Si cette comparaison trouve un LECNS de taille non-canonique supérieure à 1, le nouveau sous-graphe est ajouté au cluster. Ainsi, on s'assure que *toutes* les occurrences d'un (graphe de) motif sont bien détectées.

Les clusters de motifs potentiels ont été validés en vérifiant que :

- les motifs répertoriés dans [40, 45] (Sarcin-ricin, E-loop, K-turn et C-loop) forment bien des clusters distincts,
- les membres d'un cluster se superposent "bien" en 3D, autrement dit ont une valeur de RMSD < 1.9 Å. Cette RMSD est obtenue par invocation de la commande *align* de *Pymol* [14] qui effectue un alignement en 3D des bases tout en minimisant la RMSD. Par exemple les clusters L, M et N du dendrogramme de la figure 3.7 ne vérifient pas cette condition. Ils ne seront donc pas retenus.

La figure 3.8 montre les diagrammes 2D des motifs récurrents extraits à partir des dendrogrammes des trois structures utilisées. Les chaînes 5S de l'ARNr de *H.marismortui* et *E.coli* ne contenaient pas à elles seules de motifs récurrents.

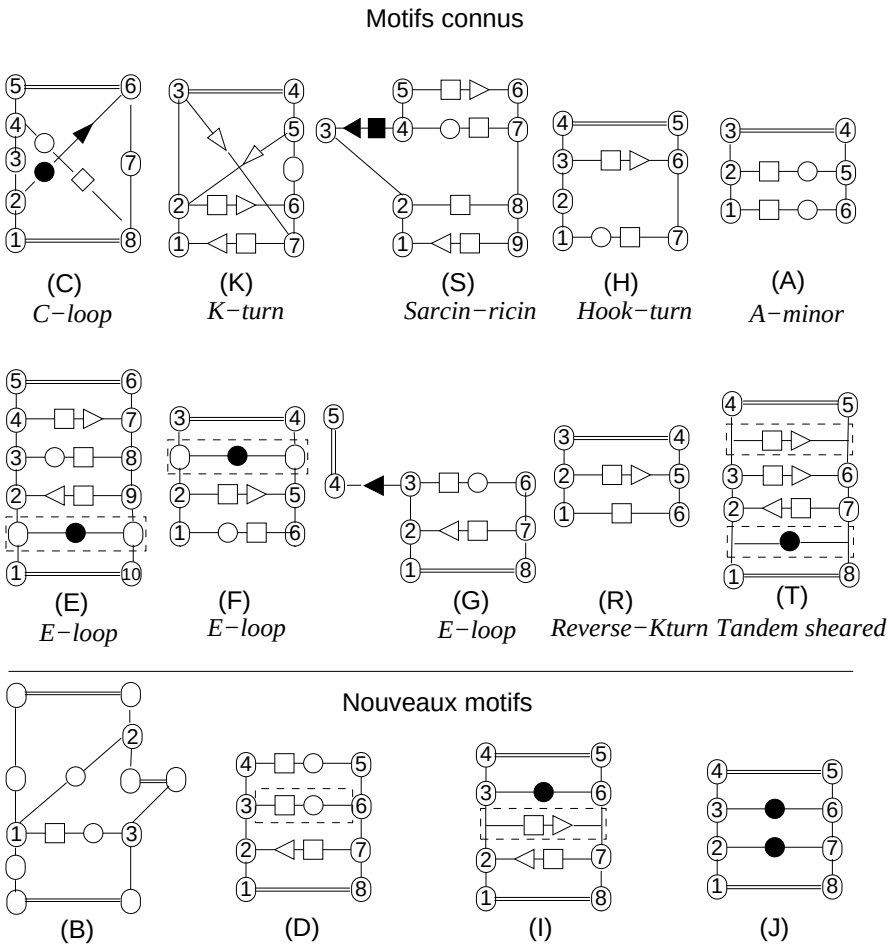


FIG. 3.8 – Diagrammes 2D des motifs récurrents trouvés dans *H.m* 23S, *E.coli* 23S et *T.th* 16S.

La table 3.1 donne pour chaque motif le nombre d'occurrences trouvées dans chaque structure. Si ce motif correspond à un motif connu, la référence bibliographique de ce dernier est également indiquée.

L'ensemble des motifs obtenus sont détaillés dans [16] et regroupés dans une page Web à l'adresse :

`http://rna3dmotif.lri.fr/repository.html`

L'annexe A montre un exemple de motif connu appelé *Hook-turn*.

Motifs	Molécule	PDB	#Occur.	Connu/Non-connu
(C)	*H.m*23S	1S72	2	C-loop [45]
	*E.coli*23S	2AW4	2	C-loop [45]
(K)	*H.m*23S	1S72	2	K-turns KT-7, KT-38 [45]
(S)	*H.m*23S	1S72	6	Sarcin-ricin [40]
	*E.coli*23S	2AW4	5	Sarcin-ricin [40]
	*T.th*16S	1J5E	2	Sarcin-ricin [40]
(H)	*H.m*23S	1S72	5	Hook-turn [65]
	*E.coli*23S	2AW4	6	Hook-turn [65]
(A)	*H.m*23S	1S72	3	A-minor [46]
(E)	*H.m*23S	1S72	3	23S E-loop [40]
	*T.th*16S	1J5E	4	23S E-loop [40]
(F)	*E.coli*23S	2AW4	5	23S E-loop dont une sarcin G2664 [40]
	*H.m*23S	1S72	5	23S E-loop dont une sarcin G911 [40]
(G)	*E.coli*23S	2AW4	2	23S E-loop [40]
(R)	*H.m*23S	1S72	7	Reverse K-turn [39]
	*E.coli*23S	2AW4	6	Reverse K-turn [39]
(T)	*E.coli*23S	2AW4	8	Tandem sheared
	*H.m*23S	1S72	6	Tandem sheared dont 2 K-turns KT-46, KT-58 [45]
	*T.th*16S	1J5E	2	Tandem sheared
(B)	*H.m*23S	1S72	2	non-connu
(D)	*E.coli*23S	2AW4	2	non-connu
(I)	*T.th*16S	1J5E	2	non-connu
(J)	*T.th*16S	1J5E	2	non-connu

TAB. 3.1 – Liste des clusters formés dans *H.m* 23S, *E.coli* 23S et *T.th* 16S.

3.4 Rna3Dmotif

La méthode décrite dans ce chapitre est implémentée dans une suite de trois programmes indépendants appelée *Rna3Dmotif* téléchargeable à l'adresse :

`http://rna3dmotif.lri.fr`

Ces programmes sont écrits en C et tournent sous Linux. Ils sont appelés *Catalog, Dendro* et *Lecns* et s'exécutent via des scripts shell.

a) Programme Catalog

Produit le catalogue des éléments de structure secondaire d'une chaîne donnée d'une structure donnée. Pratiquement, ce programme accepte en entrée le nom d'une structure "xxx.pdb" et le numéro d'ordre de la chaîne d'intérêt et produit en sortie une page HTML où les éléments de structure 2D sont listés dans une table. Chacun d'eux est donné par :

- un identifiant correspondant à son rang dans la représentation d'arbre de la structure secondaire sans pseudo-noeuds,
- l'ensemble des codes d'étiquettes de ses arêtes non-canoniques,
- un descripteur détaillant l'ensemble de ses nucléotides et les interactions qui les relient,
- une représentation 2D de son graphe produite avec *Graphviz* [4]
- une vue 3D produite avec *Pymol* [14].

Des exemples de résultats produits par ce programme sont disponibles à :

`http://rna3dmotif.lri.fr/catalogue.html`

b) Programme Dendro

Produit le dendrogramme d'un catalogue donné (fichier au format PS) par invocation de la fonction *hclust* du paquetage *R* [5].

Des exemples de dendrogrammes produits par ce programme sont téléchargeables à :

`http://rna3dmotif.lri.fr/clustering.html`

[4] http ://www.graphviz.org/
[5] http ://www.r-project.org

c) Programme Lecns

Renvoie un LECNS de deux sous-graphes correspondant à deux éléments d'un catalogue. Concrètement, prend en entrée le nom d'une structure "xxx.pdb", le numéro d'ordre de la chaîne d'intérêt ainsi que deux identifiants d'éléments du catalogue, et renvoie en sortie les mappings de leurs sommets et leurs arêtes si la taille non-canonique d'un LECNS possible est supérieure à un.

Des tests de ce programme peuvent être lancés à partir de :

`http://rna3dmotif.lri.fr/lecns.html`

Les instructions d'installation de *Rna3Dmotif* ainsi que des scénarii d'exécution sont disponibles à l'adresse :

`http://rna3dmotif.lri.fr/download.html`

Chapitre 4

Les motifs d'interaction

Dans le chapitre précédent, j'ai présenté une méthode d'extraction et de classification d'un premier type de motifs insérés dans des éléments de structure secondaire appelés *motifs locaux*. Un autre type de motifs implique des éléments de structure secondaire distincts. Le rôle de ces motifs est d'induire le repliement en des formes plus compactes de domaines [1] ou sous-domaines de la molécule. Ils font intervenir des interactions dites *longue-distance*. Le premier connu de ces motifs est le *kissing-loop* dans lequel deux boucles terminales sont reliées par un pseudo-noeud. Plus récemment, un autre motif, appelé motif *en A-mineur* a été observé de manière fréquente dans les structures d'ARN [57, 47]. Ce motif est composé d'interactions de type non-WC impliquant les côtés Sucre des bases.

Comme les motifs de ce second type font interagir différents éléments de structure secondaire, je les appellerai dans la suite *motifs d'interaction*.

Dans la section 1, je donne la définition de motif d'interaction. La section 2 montre comment on calcule la similarité de deux motifs d'interaction. Cette mesure servira à extraire les motifs d'interaction récurrents selon la méthode adoptée pour les motifs locaux. La section 3 explique une étape supplémentaire dans la méthode d'extraction. Cette étape est l'algorithme d'identification des motifs d'interaction présents dans une structure donnée. La section 4 présente les résultats obtenus.

[1] Un domaine est une région structuralement autonome d'une molécule, entièrement stabilisée par des interactions internes [54].

4.1 Définition de motif d'interaction

Les interactions longue-distance de type non-WC mènent à différents types de motifs et font intervenir deux régions en simple brin ou une région en simple brin et une hélice [72, 71]. Une région en simple brin, rappelons-le, appartient à un élément de structure secondaire (i.e. boucle interne ou terminale, renflement ou jonction).

Un exemple de motif d'interaction est le motif en A-mineur défini dans [48] comme un "un assemblage de deux adénines consecutives qui interagissent avec deux paires Watson-Crick consécutives via des interactions A-mineur" (i.e. interactions non-WC de type Sucre). (Figure 4.1)

De ce qui précède, une définition de motif d'interaction pourrait être la suivante :

> *"Un motif d'interaction est un assemblage d'interactions non Watson-Crick contiguës reliant d'un côté un élément de structure secondaire et de l'autre côté une hélice ou un autre élément de structure secondaire."*

4.2 Similarité de deux motifs d'interaction

Le calcul de la similarité de deux sous-graphes d'ARN modélisant deux motifs d'interaction potentiels nécessite le calcul de leur *plus grand sous-graphe non-canonique commun délinéable ("Largest Delineable Common Non-canonical Subgraph (LDCNS)")*.

4.2.1 Définitions

Définition 8 Soit g un sous-graphe non-canonique d'un sous-graphe d'ARN G. La **semi-complétude** de g dans G est le graphe obtenu en ajoutant à g toutes les arêtes de G de type squelette sucre-phosphate ("backbone") ayant au moins une extrémité dans g.

Définition 9 Un sous-graphe non-canonique G_{12} commun à deux sous-graphes d'ARN G_1 et G_2 est **délinéable** si ses semi-complétudes dans G_1 et G_2, respectivement, sont isomorphes.

Définition 10 Un **LDCNS** ("Largest Delineable Common Non-canonical Subgraph") de deux sous-graphes d'ARN G_1 et G_2, noté LDCNS(G_1,G_2), est

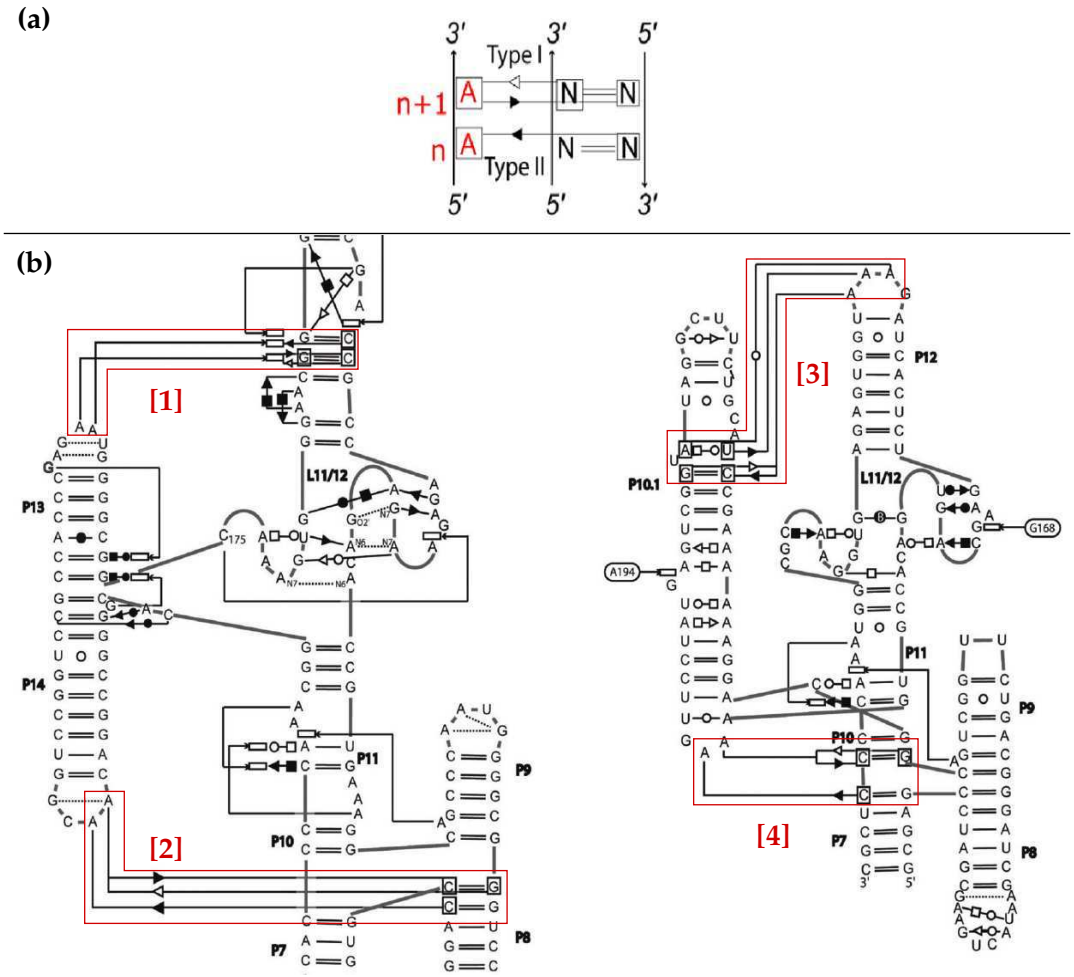


FIG. 4.1 – (a) Consensus du motif en A-mineur. Figure extraite de [47]. (b) Diagrammes des réseaux d'interaction de deux RNase P. Encadrées en rouge, des occurrences, numérotées de 1 à 4, du motif en A-mineur. Figure extraite de [46] avec adaptation.

un sous-graphe non-canonique commun délinéable de taille (non-canonique) maximum.

Note : Un LDCNS de deux sous-graphes d'ARN n'est pas nécessairement connexe.

La figure 4.2 illustre la notion de LDCNS.

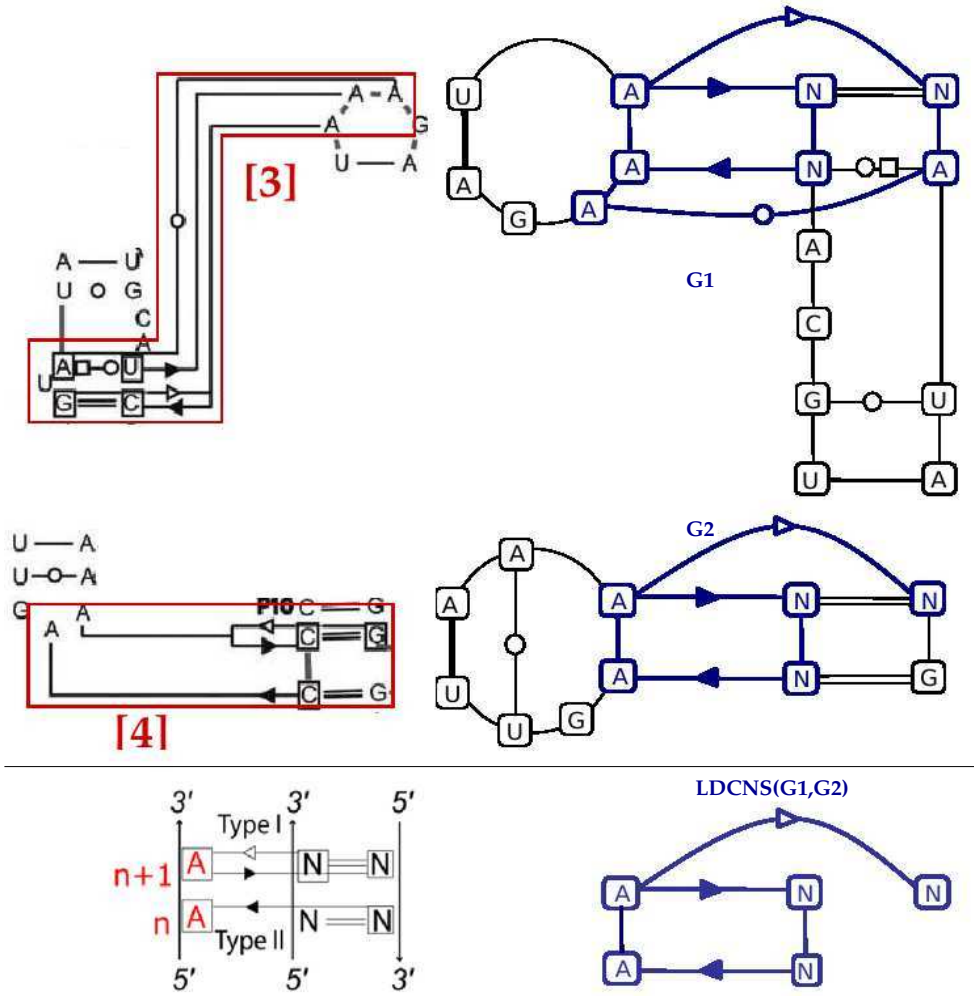


FIG. 4.2 – (Haut) En bleu, deux sous-graphes d'ARN contenant les occurrences 3 et 4 du motif en A-mineur de la figure 4.1 et de tailles non-canoniques 4 et 3 respectivement. (Bas) En bleu, un LDCNS, le seul possible, de taille non-canonique 3. Il correspond à la structure consensus du motif.

4.2.2 Mesure de similarité

La similarité de deux motifs d'interaction modélisés par deux sous-graphes d'ARN G_1 et G_2, notée $sim(G_1, G_2)$, est définie par :

$$sim(G_1, G_2) = \begin{cases} \dfrac{||LDCNS(G_1, G_2)||}{max(||G_1||, ||G_2||)} & \text{si } ||LDCNS(G_1, G_2)|| > 1 \\ 0 & \text{sinon.} \end{cases} \quad (4.1)$$

La mesure de similarité vérifie les propriétés suivantes :

- $0 \leq sim(G_1, G_2) \leq 1$,
- $sim(G_1, G_2) = sim(G_2, G_1)$ (Symétrie)
- $sim(G_1, G_2) = 1 \Longrightarrow$ les semi-complétudes des plus grands sous-graphes non-canoniques de G_1 et G_2 sont isomorphes.
- $sim(G_1, G_2) = 0 \Longrightarrow G_1$ et G_2 n'ont pas de sous-graphe non-canonique délinéable commun de taille > 1.

Exemple : La similarité des deux occurrences du A-mineur de la figure 4.2 est égale à $3/max(3, 4) = 0.75$

De manière duale, la mesure de dissimilarité de deux motifs d'interaction modélisés par deux sous-graphes d'ARN G_1 et G_2, notée $dis(G_1, G_2)$, est définie par :

$$dis(G_1, G_2) = 1 - sim(G_1, G_2)$$

4.2.3 Implémentation

On utilise la même représentation de graphe que celle vue au chapitre 3.

Le principe de fonctionnement de la procédure LDCNS est le même que celui de la procédure LECNS, à la différence près que la procédure *Complétude* (g, G) est remplacée par la procédure *Semi-complétude* (g, G). Les complexités au pire (en temps et en espace) sont donc égales à celles de la procédure LECNS.

La procédure *Semi-complétude* est identique à la procédure *Complétude* (cf. section 3.1.3) excepté que dans la ligne 5 le test du type de l'arête se limite au type "backbone". Les complexités au pire sont donc égales à celles de la procédure *Complétude*.

4.3 Extraction des motifs d'interaction

La méthode d'extraction des motifs d'interaction récurrents est similaire à la méthode d'extraction des motifs locaux récurrents(cf. section 3.2). Dans le cas des motifs d'interaction, une étape supplémentaire est ajoutée : l'identification des motifs d'interaction après la construction du catalogue des éléments de structure secondaire. Cette étape est réalisée par le module **Listing** (Figure 4.3).

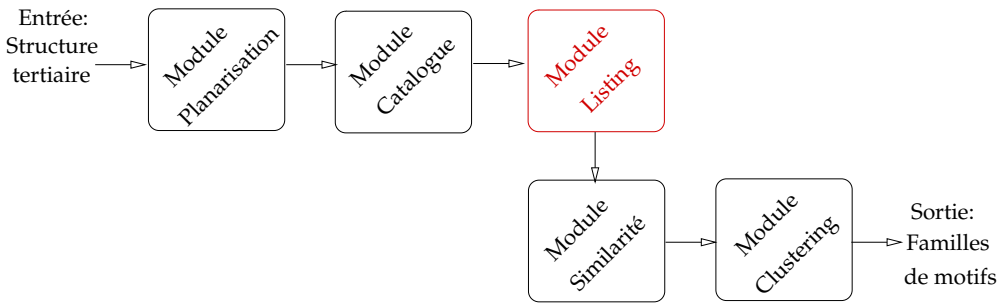


FIG. 4.3 – La méthode d'extraction des motifs d'interaction nécessite une étape supplémentaire : produire la liste des motifs d'interaction d'une structure. Cette étape est réalisée par le module Listing.

4.3.1 Données et catalogue

Un échantillon de structures cristallographiques, annotées par FR3D [61], ont été téléchargées[2] depuis l'adresse :

`http://rna.bgsu.edu/FR3D/`

Pour chacune de ces structures on construit son catalogue comme décrit dans la section 3.2.2.

4.3.2 Module Listing

Une fois le catalogue consruit, on identifie les motifs d'interaction grâce à la définition de la section 4.1 en procédant comme suit : pour chaque élément du catalogue, vérifier s'il possède des bases libres impliquées dans des interactions non-canoniques distantes, c'est à dire des interactions dont une seule extrémité appartient à l'élément considéré du catalogue. Dans ce cas, un motif d'interaction est trouvé. Il sera composé de l'ensemble de ces interactions non-canoniques distantes. Ajouter alors au motif les interactions de type empilement reliant deux quelconques des bases aux extrémités de ses interactions.

Un motif d'interaction est modélisé par un sous-graphe d'ARN. L'algorithme d'identification des motifs d'interaction d'une structure donnée est le suivant :

[2]Version du 10 Octobre 2008

Procédure Listing

Entrée : Catalogue de la structure secondaire

Sortie : Liste IM de sous-graphes d'ARN, initialement vide

---

1 Début
2 Pour chaque élément du catalogue faire
3 $BL \leftarrow$ l'ensemble des bases libres
4 $ID \leftarrow$ l'ensemble des interactions non-canoniques distantes ayant
5 une extrémité dans BL
6 Si ID n'est pas vide
7 Créer un sous-graphe $G = (V, E)$;
8 $E \leftarrow$ les éléments de ID ;
9 $V \leftarrow$ les bases aux extrémités des arêtes de E ;
10 Ajouter à E les arêtes de type empilement ayant leurs
11 deux extrémités dans V ;
12 $IM \leftarrow IM \cup \{G\}$;
13 Fin si ;
14 Fin Pour ;
15 Supprimer de IM les sous-graphes redondants ;
16 Retourner la liste IM ;
17 Fin.

La ligne 15 supprime les motifs d'interaction en double. En effet, supposons qu'un motif d'interaction m_{ij} relie deux éléments de structure secondaire appelés e_i et e_j (tels que $i < j$). Lors du parcours de la liste des éléments du catalogue (Ligne 2), on rencontre e_i en premier. L'exécution des lignes 3 à 13 ajoute le sous-graphe correspondant à m_{ij} à la liste IM. On poursuit le parcours des éléments du catalogue. On rencontre e_j. L'exécution des lignes 3 à 13 ajoutera le sous-graphe correspondant à m_{ij} à la liste IM. Comme il s'agit du même motif, une seule "copie" sera gardée.

Dans la suite, j'appellerai :

- **Interacting interfaces** les éléments de structure secondaire et les hélices auxquels appartiennent les bases aux extrémités des arêtes d'interaction,
- **Contexte structural** l'ensemble de ces interacting interfaces.

Des exemples de listings de structures d'ARN sont disponibles à l'adresse :

`http://www.lri.fr/~md/RIM/bychain.html`

(colonne du tableau intitulée "Listing of interaction motifs")

4.3.3 Similarité

Pour chaque paire de motifs d'interaction, pouvant appartenir soit à la même structure soit à des structures différentes, on calcule une mesure de dissimilarité en utilisant la formule de la section 4.2.2. Puis, on remplit une matrice de dissimilarités sur laquelle on applique un clustering hiérarchique en utilisant la méthode UPGMA. Pratiquement, ceci est réalisé en invoquant la fonction *hclust* du paquetage *R project for statistical computing*.[3]

4.3.4 Clustering

Cette étape de clustering produit un dendrogramme. Les motifs récurrents sont alors extraits à partir d'un seuil de similarité optimal de 1 (i.e. dissimilarité égale à 0). Autrememt dit, les clusters sont formés de (sous-graphes de) motifs d'interaction *rigoureusement identiques*.

Ce seuil optimal a été déduit à partir de l'observation suivante : le motif en A-mineur de la figure 4.1 est un assemblage de deux interactions Sucre appelées respectivement *Type I* et *Type II* (Figure 4.4) et est, par conséquent, appelé A-mineur Type I/Type II.

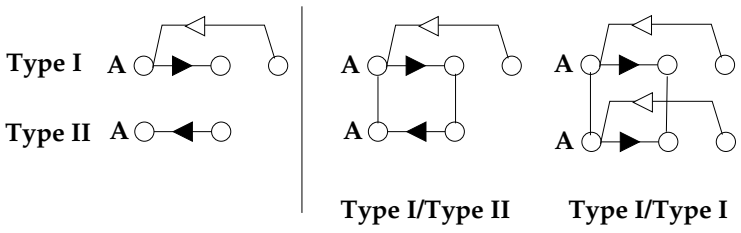


FIG. 4.4 – Différentes combinaisons des interactions A-mineur Type I et Type II forment différentes familles de motif en A-mineur.

Des variantes distinctes du A-mineur peuvent être obtenues par différentes combinaisons de ces deux types d'interaction. Ainsi, un A-mineur Type I/ Type I est une famille différente du A-mineur Type I/ Type II. Or, vu leurs modèles de graphe dans lesquels l'arête *cis* Sucre/Sucre est considérée comme

[3]http ://www.r-project.org/

Molécule	Organisme	PDB	Chaîne	Résolution (Å)
ARNt ASP	*Yeast*	2TRA	A	0.00
ARNt PHE	*Yeast*	1EHZ	A	0.00
Groupe I intron	*Azoarcus*	1U6B	B	3.10
Groupe II intron	*Synthetic construct*	3EOH	A	0.00
RNaseP Type A	*T.thermophilus*	1U9S	A	2.90
	T.maritima	2A2E	A	3.85
RNaseP Type B	*B.subtilis*	1NBS	A,B	3.15
	B.stearothermophilus	2A64	A	3.30
16S ARNr	*T.thermophilus*	1J5E	A	3.05
	E.coli	2AVY	A	3.46
23S ARNr	*H.marismortui*	1S72	0	2.40
	E.coli	2AW4	B	3.46
	T.thermophilus	2HGU	A	4.50
	D. radiodurans	1NKW	0	3.10
5S ARNr	*H.marismortui*	1S72	9	2.40
	E.coli	2AW4	A	3.46
	T.thermophilus	2HGU	B	4.50
	D.radiodurans	1NKW	9	3.10
Hammerhead ribozyme	*S.mansoi*	2GOZ	A	2.20
	Synthetic construct	2OEU	A	2.00
Guanine riboswitch	*B.subtilis*	1U8D	A	1.95
		1Y27	X	2.40
	V.vulnificus	1Y26	X	2.10
SAM riboswitch	*T.tengcongensis*	2GIS	A	2.90
TPP riboswitch	*E.coli*	2GDI	X,Y	2.05
	Arabidopsis	2CKY	A.B	2.90

TAB. 4.1 – Liste des structures cristallographiques utilisées.

symétrique, ces deux motifs possèdent une similarité de 3/4 ce qui correspond à une dissimilarité de 0.25. Donc, si nous voulons que ces deux familles soient regroupées dans des clusters séparés, il faut couper l'arbre du dendrogramme à un seuil de dissimilarité inférieur à 0.25. D'où le seuil 0.

4.4 Résultats

Le pipeline d'extraction a été appliqué à l'ensemble des structures de la table 4.1.

Les motifs obtenus à la fin du clustering ont été validés par superposition en 3D en utilisant la commande *align* de *Pymol* [14] (une RMSD < 1.69 Å indiquait une "bonne" superposition). Ces motifs sont au nombre de treize et

Motif	#Occur.	Connu/Nouveau
IM1	15	A-mineur Type I/ Type II
IM2	3	variante A-mineur
IM3	4	variante A-mineur
IM4	4	variante A-mineur
IM5	2	Nouveau
IM6	4	Nouveau
IM7	3	Nouveau
IM8	2	Nouveau
IM9	2	Nouveau
IM10	2	Nouveau
IM11	2	Nouveau
IM12	2	Nouveau
IM13	2	Nouveau

TAB. 4.2 – Motifs d'interaction récurrents trouvés dans les structures de la table 4.1.

sont regroupés dans la table 4.2.

La figure 4.5 montre les diagrammes 2D de quelques uns des 13 motifs d'interaction trouvés. Ces diagrammes font apparaître le motif en bleu dans son contexte structural. La liste de tous les motifs d'interaction est disponible à l'adresse :

`http://www.lri.fr/~md/RIM/bysim.html`

L'annexe B donne les détails d'un motif d'interaction nouveau appelé IM6.

4.5 Comme des pièces d'un puzzle

Nous avons vu dans le premier chapitre qu'un motif (local) d'ARN est défini comme un ensemble de paires de bases non-Watson-Crick isostériques ordonnées et orientées qui mènent à un repliement caractéristique du squelette sucre-phosphate (cf. section 1.1).

De même, dans la section 4.1, j'ai défini un motif d'interaction comme un assemblage d'interactions non-Watson-Crick contiguës reliant d'un côté un élément de structure secondaire et de l'autre côté une hélice ou un autre élément de structure secondaire.

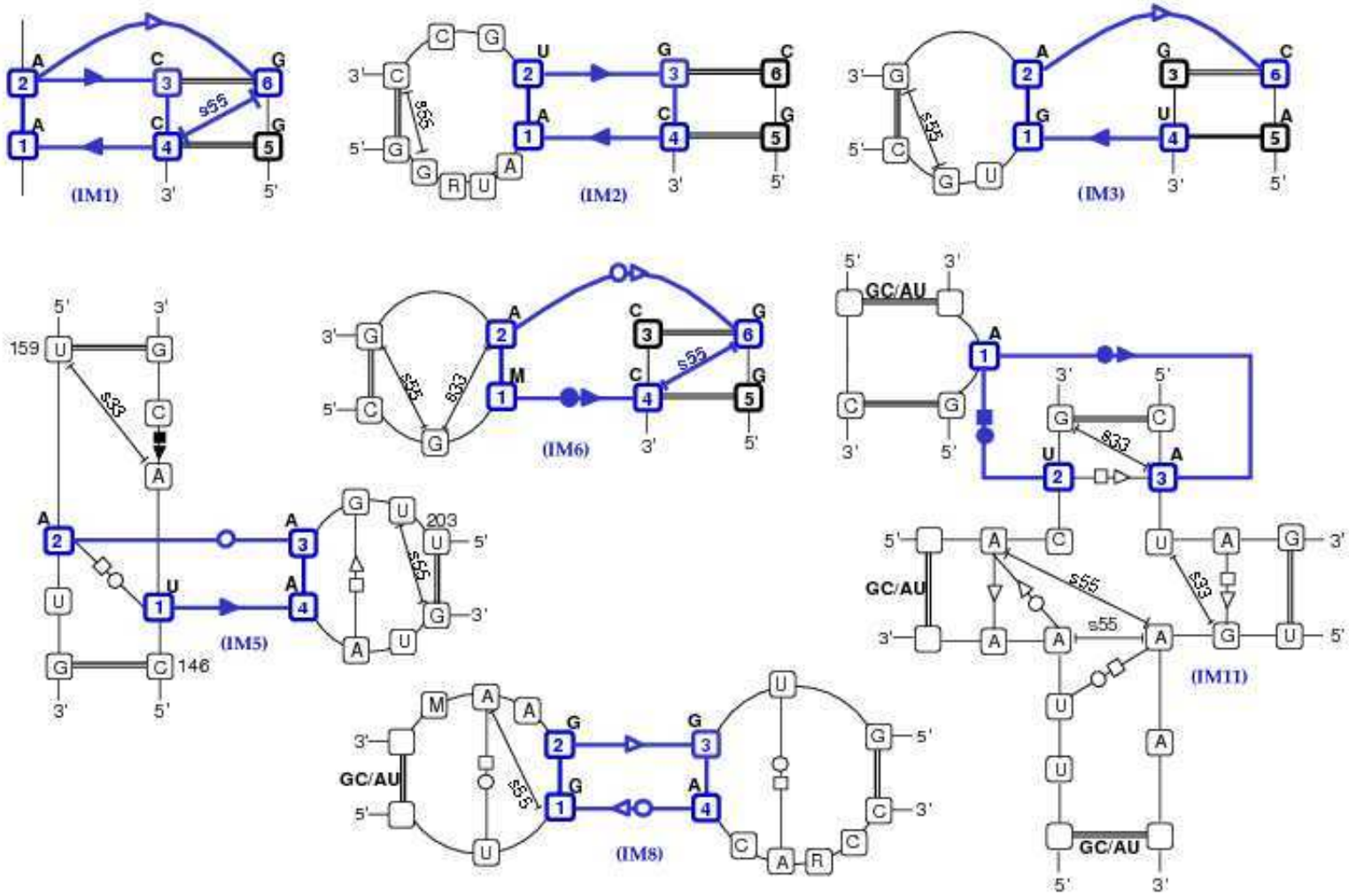


FIG. 4.5 – Exemples de motifs d'interaction récurrents mis en évidence en bleu dans leur contexte structural.

Ces deux définitions confirment une intuition universellement admise par la communauté structuraliste. Cette intuition voit les motifs d'ARN comme des poupées russes pouvant s'emboîter les unes dans les autres pour former des motifs de tailles variables. Par exemple, dans les motifs locaux de la figure 3.8 (section 3.3) le motif F qui est une variante de la boucle E est un sous-motif du motif S (i.e. Sarcin-ricin). Aussi, le motif (T) qui est le bien connu tandem sheared est un sous motif du motif K (i.e. Kink-turn). De même, dans les motifs d'interaction de la figure 4.5, nous pouvons voir que les motifs IM2 et IM3 sont des sous-motifs de IM1 (le motif en A-mineur).

La question qui se pose naturellement est la suivante : quelle est la *plus petite* brique qui compose un motif structural ?

La réponse à cette question n'a pas encore été tranchée. La raison est qu'il n'existe pas encore de vocabulaire consensuel pour décrire une structure d'ARN et par extension ses motifs. D'ailleurs, c'est l'un des objectifs immédiats du Consortium de l'Ontologie de l'ARN (ROC) [3, 38] [4] que de définir de manière précise les concepts permettant aux structuralistes de manipuler un langage commun.

Une première proposition de réponse me semble, néanmoins, possible si l'on réduit la granularité d'un motif à sa plus petite composante : l'interaction non-Watson-Crick. Ainsi, un motif peut être vu comme un puzzle dont les pièces élémentaires sont les différentes interactions non-WC qui le composent.

Par exemple, le motif en A-mineur est un puzzle composé de deux pièces : une interaction A-mineur Type II et une interaction A-mineur Type I. On notera ici qu'un motif A-mineur Type I, bien que composé de deux interactions Sucre, est compté comme une seule interaction. Plus généralement, "si une base est reliée par deux interactions non-WC à deux bases elles mêmes reliées par une paire de bases WC alors les deux interactions non-WC comptent pour une seule." (Communication personnelle de E. Westhof).

Dans une tentative de déceler les interactions tertiaires qui composent fréquemment les motifs d'interaction ainsi que leurs partenaires préférentiels, j'ai décomposé les 13 motifs récurrents de la table 4.2 en leurs pièces élémentaires : les interactions tertiaires. Ensuite, j'ai organisé ces interactions sous la forme d'un tableau dans lequel on peut voir pour chaque interaction tertiaire, son interaction (tertiaire) partenaire et dans quel motif elles sont

[4] http ://roc.bgsu.edu/

combinées.

Ce tableau est disponible à l'adresse :

`http://www.lri.fr/~md/RIM/puzzles.html`

La figure 4.6 montre un exemple d'interaction tertiaire, *trans* WC/WC, qui contribue à la composition de deux motifs d'interaction différents : IM5 et IM12.

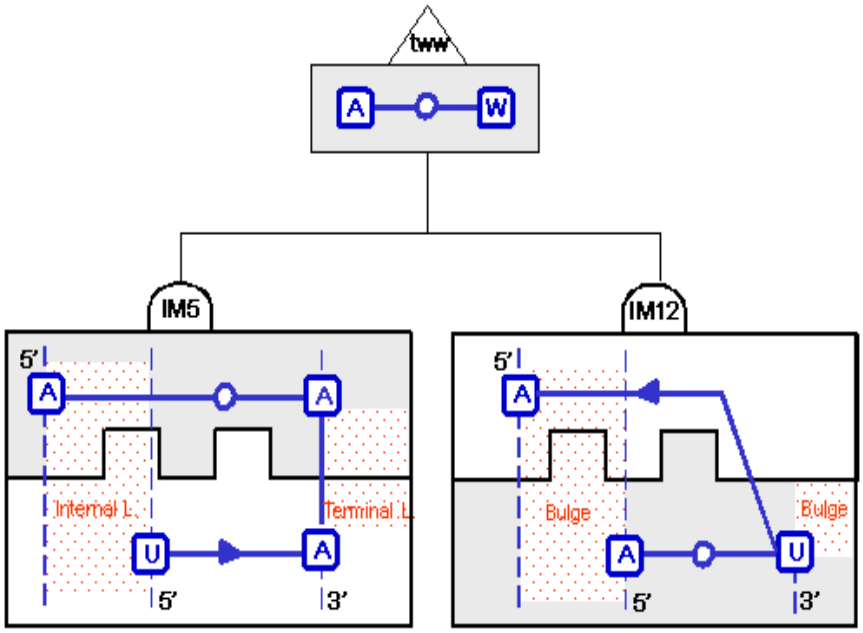


FIG. 4.6 – Motifs d'interaction vus comme des combinaisons variées des pièces d'un puzzle, i.e. les interactions tertiaires. Dans cet exemple, les mêmes pièces ou interactions sont assemblées de deux manières différentes pour former deux motifs différents.

Il est remarquable de constater que cette interaction *trans* WC/WC est toujours associée au même partenaire : une interaction *cis* Sucre/Sucre. Pour savoir si cette corrélation n'est pas limitée à l'échantillon de structures considéré, la méthode d'extraction devra être appliquée à l'ensemble des structures non redondantes connues, par exemple celles disponibles à l'adresse :

`http://rna.bgsu.edu/FR3D/AnalyzedStructures/NonredundantList.txt`

Conclusion et perspectives

Afin d'être biochimiquement actives, les molécules d'ARN adoptent des formes tridimensionnelles complexes. Cette structuration en 3D est assurée par les motifs tertiaires dont la présence récurrente, indépendante du contexte structural, démontre l'importance de leur rôle architectural. L'étude de ces motifs structuraux consitue donc une voie prometteuse vers une meilleure compréhension des fonctions biologiques des molécules d'ARN.

Les méthodes automatiques d'identification des motifs tertiaires développées à ce jour présentent l'inconvénient de considérer comme rigoureusement identiques les différentes occurrences des motifs recherchés. Or, dans la réalité, ces occurrences sont souvent similaires à quelques insertions près de bases libres ou de paires de bases.

Dans ma thèse, j'ai proposé une méthode globale d'extraction automatique de motifs tertiaires d'ARN dont les occurrences sont similaires. Cette méthode utilise un modèle de graphe de la structure tertiaire et repose sur une notion de similarité de graphe qui exploite de manière optimale la définition structurale de motif tertiaire.

Le pour et le contre

Les résultats obtenus ont montré que cette méthode est fiable. En effet, les motifs trouvés vérifient la définition de motif structural et sont validés par superposition en 3D. La méthode a découvert de nouveaux motifs dans des structures connues. La notion de similarité de graphe proposée est flexible puisqu'il est facile de l'adapter à deux types différents de motifs tertiaires.

La qualité des résultats est, toutefois, sensible à la précision des programmes d'annotation qui produisent le graphe d'ARN. Plus l'annotation est précise, plus le modèle de graphe est fidèle à la structure tertiaire qu'il représente. Comme il existe trois programmes d'annotation (*Rnaview*[74], *FR3D*[61] et

MC-annotate[36]), il serait utile de réaliser un compratif de leurs qualités d'annotation (voir point **(b)** plus bas) pour obtenir les meilleurs résultats possibles.

Prolongements envisagés

Au terme de trois années de travail, certaines étapes du pipeline de la méthode s'avèrent améliorables.

I) Les données

(a) Appliquer la méthode à l'ensemble des structures non redondantes

Dans mes tests j'ai utilisé de petits échantillons de structures. L'objectif initial était de vérifier la fiabilité de la méthode plutôt que découvrir de nouveaux motifs. Cette validation étant faite, je propose d'appliquer la méthode à l'ensemble des structures non redondantes pour identifier d'éventuels nouveaux motifs.

II) La méthode

(b) Possibilité de choisir un programme d'annotation

Dans mon implémentation, j'ai utilisé deux programmes d'annotation différents : *Rnaview*[74] dans la partie conscarée aux motifs locaux et *FR3D* [61] dans la partie dédiée aux motifs d'interaction. Il serait, cependant, intéressant d'offrir à l'utilisateur la possibilité de choisir l'annotateur en fonction de la qualité de sa précision. On notera qu'il existe un troisième programme d'annotation *MC-annotate* [36]. Si cette interface est développée, il faudra uniformiser le format de sortie du graphe produit par les différents programmes.

(c) Automatiser l'extraction des clusters

Le module **Clustering** fait appel à la fonction *hclust* du paquetage *R*. Cette fonction utilise la méthode UPGMA et produit une image (i.e. arbre de classification) sous la forme d'un fichier au format PS. Sur cette image, le seuil de dissimilarité optimal est tracé à la main pour extraire les clusters correspondant aux motifs potentiels. Cette étape peut être automatisée en reprogrammant l'algorithme UPGMA et l'intégrant au pipeline de la méthode de sorte que le résultat du module **Clustering** ne soit plus un arbre de classification mais une liste de clusters formés par le seuil optimal et contenant les

occurrences des motifs potentiels.

(d) Automatiser l'extraction de la structure consensus

La structure consensus des motifs trouvés est extraite en effectuant à la main deux étapes :(1) construction du noyau non-canonique par alignement multiple des mappings appariés ("pairwise mappings"). Ces mappings appariés sont ceux que le programme LECNS/LDCNS renvoie lorsqu'il est exécuté sur une paire de sous-graphes du cluster, (2) extension du noyau non-canonique aux paires de bases WC flanquant le motif en autorisant l'insertion d'un nombre "modéré" de bases libres ou de paires de bases. Il est possible d'automatiser ces deux étapes de la manière suivante : (1) la construction du noyau non-canonique peut se faire en définissant une structure de donnée adéquate qui stockerait l'alignement multiple des différents mappings appariés au fur et à mesure qu'ils sont produits par les appels au programme LECNS/LDCNS, (2) l'extension du noyau non-canonique aux paires de base WC flanquant le motif peut être réalisée en formulant de manière précise les règles implicites adoptées par les biochimistes. Mon opinion est que l'on est autorisé à insérer un nombre quelconque de bases libres mais pas plus d'une paire de bases non-WC entre le noyau non-canonique et la paire de base WC qui le flanque.

(e) Automatiser la validation des motifs trouvés par superposition en 3D

L'ultime étape de validation des occurrences des motifs trouvés consiste à effectuer une superposition en 3D en utilisant un visonneur de molécules (dans mon cas, la commande *align* de *Pymol* [14]). Le constat d'une bonne superposition se fait visuellement. Cette étape peut être automatisée en utilisant la nouvelle version des matrices d'isostérie définies dans un travail récent de Stombaugh et collaborateurs [64]. Pratiquement, il s'agira d'effectuer des tests pendant la réalisation de l'alignement multiple : pour deux arêtes (appartenant à deux occurrences différentes) associées par le programme LECNS/LDCNS, vérifier si elles sont isostériques. Si oui, poursuivre l'alignement. Si non, une des deux occurrences devra être écartée.

(f) Décomposition en pièces de puzzle

Appliquer la décomposition en briques élémentaires aux motifs trouvés, notamment les motifs locaux. Pour ce type de motifs, il serait plus indiqué d'augmenter la granularité (par exemple choisir comme brique de base un assemblage de deux interactions tertiaires plutôt qu'une). Le but étant de découvrir d'éventuelles relations d'emboîtement entre familles de motifs à la

manière des poupées russes.

III) Les résultats

(g) Automatiser le dessin des diagrammes 2D des motifs trouvés

C'est une tâche très utile mais qui n'a pas encore reçu l'attention qu'elle mérite en dépit d'une forte demande de la part des communautés biochimiste et bioinformaticenne.

(h) Construire une base de données avec les motifs trouvés

En vue de l'intégrer à une plateforme de modélisation de structure 3D comme, par exemple, *Assemble*[5].

Feuille de route

Un plan possible de mise en oeuvre de ces améliorations (classées par odre de priorité) serait le suivant :

Court terme

1. **(c)** Automatiser l'extraction des clusters.
2. **(d)** Automatiser l'extraction de la structure consensus.
3. **(e)** Automatiser la validation des motifs trouvés par superposition 3D.
4. **(g)** Automatiser le dessin des diagrammes 2D des motifs trouvés.

Moyen terme

5. **(b)** Possibilité de choisir un programme d'annotation.
6. **(a)** Appliquer la méthode à l'ensemble des structures non redondantes.
7. **(h)** Construire une base de donnés avec les motifs trouvés.

Long terme

8. **(f)** Décomposition en pièces de puzzle.

[5]http ://www.bioinformatics.org/assemble/

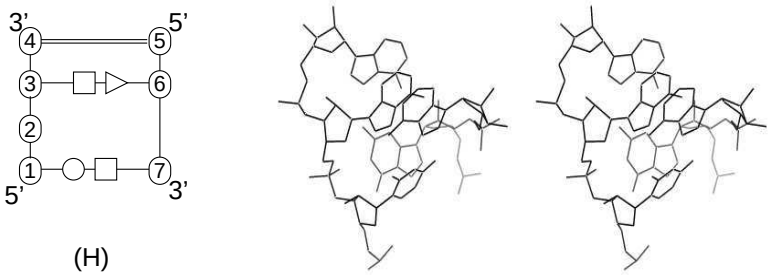
Annexes

Annexe A

Exemple de motif local connu, le *Hook-turn* (cluster H), donné par :

- le diagramme 2D de sa structure consensus,
- la vue stréréo de son occurrence référence,
- et sa séquence signature.

Pour plus de détails, voir [16].



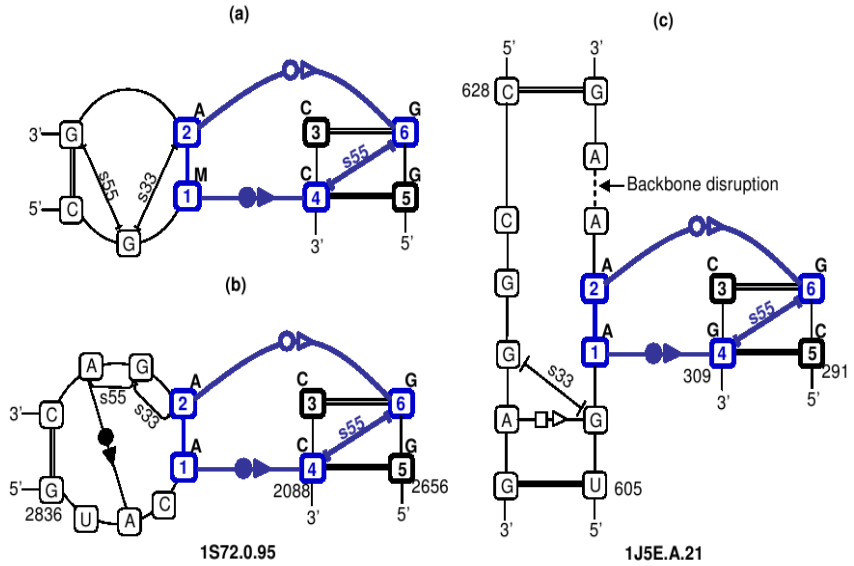
(H)

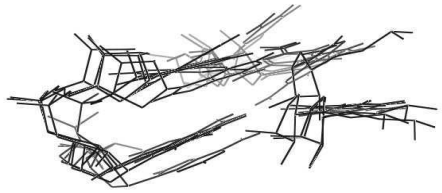
PDB	Id.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Catalogue	RMSD (ref.H83)
1S72	H83	1096_U	1097_A	1098_A	1099_G	1257_C	1258_G	1259_A	Internal L.(83)	0.00
	H111	1457_U	1458_A	1459_A	1460_G	1483_C	1484_G	1485_A	Internal L.(111)	0.76
	H201	2774_U	2775_A	2776_A	2777_G	2797_C	2798_G	2799_A	Internal L.(201)	0.33
	H161	2242_U	2243_C	2244_A	2245_C	2256_G	2257_G	2258_A	Junction L.(161)	1.61
	H193	2673_U	2674_G	2675_A	2676_C	2809_G	2810_G	2811_A	Internal L.(193)	1.61
2AW4	H73	999_U	1000_A	1001_A	1002_G	1153_C	1154_G	1155_A	Internal L.(73)	0.42
	H101	1352_U	1353_A	1354_A	1355_G	1376_C	1377_G	1378_A	Internal L.(101)	0.69
	H106	1578_U	1579_A	1580_A	1581_G	1417_C	1418_G	1419_A	Internal L.(106)	0.31
	H205	2739_U	2740_A	2741_A	2742_G	2762_C	2763_G	2764_A	Internal L.(205)	0.52
	H161bis	2197_U	2198_A	2199_A	2200_C	2223_G	2224_G	2225_A	Junction L. (161)	1.60
	H196	2637_U	2638_G	2639_A	2640_G	2774_C	2775_G	2776_A	Internal L.(196)	1.66

Annexe B

Exemple de motif d'interaction appelé IM6 donné par :
- les diagrammes 2D des contextes structuraux de ses occurrences,
- la superposition en 3D de ses occurrences,
- et sa séquence signature.

L'occurrence référence dans la superposition 3D est repérable par une RMSD nulle.





Molecule	Organism	PDB-Chain	Identifier	(1)	(2)	(3)	(4)
23S	*E.coli*	2AW4-B	2AW4.B.45	1754_A	1755_A	2715_C	2716_C
23S	*H.m*	1S72-0	1S72.0.66	1810_C	1811_A	2751_C	2752_C
23S	*H.m*	1S72-0	1S72.0.95	2840_A	2841_A	2087_C	2088_C
16S	*T.th*	1J5E-A	1J5E.A.21	607_A	608_A	308_C	309_G

(5)	(6)	Diagram2D	Interacting bases	RMSD (Å)
2693_G	2694_G	(a) png	pdb	0.000
2730_G	2731_G	(a) png	pdb	0.861
2756_G	2757_G	(b) png	pdb	0.221
291_C	291_G	(c) png	pdb	0.627

Bibliographie

[1] J. Allali. *Comparaison de structures secondaires d'ARN*. PhD thesis, Université de Marne-la-Vallée, Décembre 2004.

[2] P.J. Artymiuk, R.V. Spriggs, and P. Willett. Graph theoretic methods for the analysis of structural relationships in biological macromolecules : Research Articles. *Journal of the American Society for Information and Technology*, 56(5) :518–528, 2005.

[3] C. Batchelor, T. Bittner, K. Eilbeck, C. Mungall, J. Richardson, R. Knight, J. Stombaugh, C. Zirbel, E. Westhof, and N.B. Leontis. The RNA Ontology (RNAO) : An ontology for integrating RNA sequence and structure data. In *Proceedings of the International Conference on Biomedical Ontology*, Buffalo, NY, 2009. Available from Nature Precedings <http ://hdl.handle.net/10101/npre.2009.3561.1>.

[4] R.T. Batey, R.P. Rambo, and J.A. Doudna. Tertiary motifs in RNA structure and folding. *Angew. Chem. Int. Ed.*, 32 :2326–2343, 1999.

[5] E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunication, Paris, France, 2002.

[6] H.M. Berman, W.K. Olson, D.L. Beveridge, J. Westbrook, A. Gelbin, T. Demeny, S.-H.Hsieh, A. R. Srinivasan, and B. Schneider. The Nucleic Acid Database : A Comprehensive Relational Database of Three-Dimensional Structures of Nucleic Acids. *Biophysical Journal*, 63 :751–759, 1992.

[7] C. Bron and J. Kerbosch. Algorithm 457 : finding all cliques of an undirected graph. *Communications of the ACM*, 16(9) :575–577, 1973.

[8] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8) :689–694, 1997.

[9] H. Bunke. Error Correcting Graph Matching : On the Influence of the Underlying Cost Function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9) :917–922, 1999.

[10] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4) :255–259, 1998.

[11] J. Chen, I.A. Kanj, and W. Jia. Vertex cover : further observations and further improvements. *Journal of Algorithms*, 41(2) :280–301, 2001.

[12] D. Conte, P. Foggia, and M. Vento. Challenging Complexity of Maximum Common Subgraph Detection Algorithms : A Performance Analysis of Three Algorithms on a Wide Database of Graphs. *Journal of Graph Algorithms and Applications*, 11(1) :99–143, 2007.

[13] B. Cuissart. *Plus grande structure commune à deux graphes : méthode de calcul et intérêt dans un contexte SAR*. PhD thesis, Université de Caen/Basse-Normandie, Décembre 2004.

[14] W.L. DeLano. The pymol molecular graphics system. *DeLano Scientific, Palo Alto, CA, USA. http ://www.pymol.org*, 2002.

[15] E. Deza and M.M. Deza. *Encyclopedia of Distances*. Springer, 2009.

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

[17] J.A. Doudna. Structural genomics of RNA. *Nature Structural Biology*, 7(11) :954–956, 2000.

[18] R.G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[19] C.M. Duarte, L.M. Wadley, and A.M. Pyle. RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space. *Nucleic Acids Research*, 31(16) :4755–4761, 2003.

[20] P. J. Durand, R. Pasari, J. W. Baker, and C. Tsai. An efficient algorithm for similarity analysis of molecules . *Internet Journal of Chemistry*, 2, 1999.

[21] P. Foggia, C. Sansone, and M. Vento. A Database of Graphs for Isomorphism and Sub-Graph Isomorphism Benchmarking. In *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, pages 176–188, Ischia, Italy, May, 2001.

[22] M.R. Garey and D.S. Johnson. *Computers and intractability*. Freeman, 1979.

[23] S. Griffiths-Jones, S. Moxon, M. Marshall, Ajay A. Khanna, S.R. Eddy, and A. Bateman. Rfam : annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, 33 :D121–124, 2005.

[24] A. Harrison, D.R. South, P. Willett, and P.J. Artymiuk. Representation, searching and discovery of patterns of bases in complex RNA structures. *Journal of Computer-Aided Molecular Design*, 17(8) :537–549, 2003.

[25] D.K. Hendrix, S.E. Brenner, and S.R. Holbrook. RNA structural motifs : building blocks of a modular molecule . *Quarterly Reviews of Biophysics*, 38 :221–243, 2005.

[26] E. Hershkovitz, E. Tannenbaum, S.B. Howerton, A. Sheth, A. Tannenbaum, and L.D. Williams. Automated identification of RNA conformational motifs : theory and application to the HM LSU 23S rRNA. *Nucleic Acids Research*, 31 :6249–6257, 2003.

[27] D. Hidovic and M. Pelillo. Metrics For Attributed Graphs Based On The Maximal Similarity Common Subgraph. *IJPRAI*, 18(3) :299–313, 2004.

[28] M. Höchsmann, T. Töller, R. Giegerich, and S. Kurtz. Local Similarity in RNA Secondary Structures. In *CSB '03 : Proceedings of the IEEE Computer Society Conference on Bioinformatics*, volume 2, pages 159–168, Washington, DC, USA, 2003. IEEE Computer Society.

[29] S.R. Holbrook. RNA structure : the long and the short of it. *Current Opinion in Structural Biology*, 15 :302–308, 2005.

[30] J. A. Holland, M. R. Hansen, Z. Du, and D.W. Hoffman. An examination of coaxial stacking of helical stems in a pseudoknot motif : the gene 32 messenger RNA pseudoknot of bacteriophage T2 . *RNA* , 5 :257–271, 1999.

[31] H.C. Huang, U. Nagaswamy, and G.E. Fox. The application of cluster analysis in the intercomparison of loop structures in RNA. *RNA*, 11(4) :412–423, 2005.

[32] X. Huang, J. Lai, and S. F. Jennings. Maximum common subgraph : some upper bound and lower bound results. *BMC Bioinformatics*, 7(Suppl 4) :S6, 2006.

[33] V. Kann. *On the Approximability of NP-complete Optimization Problems*. PhD thesis, Royal Institute of Technology, S-100 44 Stockholm, May 1992.

[34] V. Kann. On the Approximability of the Maximum Common Subgraph Problem. In *Lecture Notes In Computer Science, Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science*, volume 577, pages 377–388. Springer-Verlag, 1992.

[35] C.W.H. Lam and L.H. Soicher. Three new combination algorithms with the minimal change property. *Commununications of the ACM*, 25(8) :555–559, 1982.

[36] S. Lemieux and F. Major. RNA canonical and non-canonical base pairing types : a recognition method and complete repertoire. *Nucleic Acids Research*, 30(19) :4250–4263, 2002.

[37] S. Lemieux and F. Major. Automated extraction and classification of RNA tertiary structure cyclic motifs. *Nucleic Acids Research*, 34(8) :2340–2346, 2006.

[38] N.B. Leontis, R.B. Altman, H.M. Berman, S.E. Brenner, J.W. Brown, D.R. Engelke, S.C. Harvey, S.R. Holbrook, F. Jossinet, S.E. Lewis, F. Major, D.H. Mathews, J.S. Richardson, J.R. Williamson, and E. Westhof. The RNA Ontology Consortium : an open invitation to the RNA community. *RNA*, 12 :531–541, 2006.

[39] N.B. Leontis, A. Lescoute, and E. Westhof. The building blocks and motifs of RNA architecture. *Current Opinion in Structural Biology*, 16 :1–9, 2006.

[40] N.B. Leontis, J. Stombaugh, and E. Westhof. Motif prediction in ribosomal RNAs - Lessons and prospects for automated motif prediction in homologous RNA molecules. *Biochimie*, 84 :961–973, 2002.

[41] N.B. Leontis, J. Stombaugh, and E. Westhof. Survey and summary. The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic Acids Research*, 30 :3497–3531, 2002.

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

[43] N.B. Leontis and E. Westhof. The annotation of RNA motifs. *Comparative and Functional Genomics*, 3 :518–524, 2002.

[44] N.B. Leontis and E. Westhof. Analysis of RNA motifs. *Current Opinion in Structural Biology*, 13 :300–308, 2003.

[45] A. Lescoute, N.B Leontis, C. Massire, and E. Westhof. Recurrent structural RNA motifs, Isostericity matrices and sequence alignments. *Nucleic Acids Research*, 33 :2395–2409, 2005.

[46] A. Lescoute and E. Westhof. Survey and summary. The interaction networks of structured RNAs. *Nucleic Acids Research*, 34 :6587–6604, 2006.

[47] A. Lescoute and E. Westhof. The A-minor motifs in the decoding recognition process. *Biochimie*, 88 :993–999, 2006.

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

[49] A. Lescoute-Philipps. *Extraction de contraintes structurales à partir de comparaisons de séquences et de structures tridimensionnelles d'ARN*. PhD thesis, Université Louis Pasteur de Strasbourg, Février 2006.

[50] J.J. McGregor. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. *Software Practice and Experience*, 12(1) :23–34, 1982.

[51] K. Mehlhorn and St. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[52] V. Monev. Introduction to Similarity Searching in Chemistry. Institute of Organic Chemistry. In *Bulgarian Academy of Sciences, Sofia 1113, Bulgaria. Match-Communications in Mathematical and in Computer Chemistry 51*, pages 7–38, 2004.

[53] P.B. Moore. Structural motifs in RNA. *Annual Review of Biochemistry*, 68 :287–300, 1999.

[54] P.B. Moore. *The RNA folding problem. In The RNA world*, pages 381–401. Cold Spring Harbor, CSH Laboratory Press, New york, second edition, 1999.

[55] P.B. Moore and T.A. Steitz. The structural basis of large ribosomal subunit function. *Annual Review of Biochemistry*, 72 :813–850, 2003.

[56] L. Nasalean, J. Stombaugh, C. L. Zirbel, and N.B. Leontis. *Non-Protein coding RNAs*, volume 13 of *Springer Series in Biophysics*, chapter 1, pages 1–26. Springer Berlin Heidelberg, 2009.

[57] P. Nissen, J.A. Ippolito, N. Ban, P.B. Moore, and T.A. Steitz. RNA tertiary interactions in the large ribosomal subunit : the A-minor motif. *PNAS*, 98 :4899–4903, 2001.

[58] D. Oranit, R. Nussinov, and H. Wolfson. ARTS : alignment of RNA tertiary structures. *Bioinformatics*, 21(suppl 2) :ii47–53, 2005.

[59] Y. Ponty. *Modélisation de séquences génomiques structurées, génération aléatoire et application*. PhD thesis, Université Paris-Sud 11, Novembre 2006.

[60] J.W. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16(7) :521–533, 2002.

[61] M. Sarver, C.L. Zirbel, J. Stombaugh, A. Mokdad, and N.B. Leontis. FR3D : Finding Local and Composite Recurrent Structural Motifs in RNA 3D Structures. *Journal of Mathematical Biology*, 56(1-2) :215–252, 2008.

[62] J. Singler. Graph Isomorphism Implementation in LEDA 5.1. Technical report, Algorithmic Solutions Software GmbH, 2005.

[63] P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy*. Freeman, San Francisco, 1973.

[64] J. Stombaugh, C.L. Zirbel, E. Westhof, and N.B. Leontis. Frequency and isostericity of RNA base pairs. *Nucleic Acids Research*, 37(7) :2294–2312, 2009.

[65] S. Szep, J. Wang, and P.B. Moore. The crystal structure of a 26-nucleotide RNA containing a hook-turn. *RNA*, 9 :44–51, 2003.

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

[67] G. Valiente. *Algorithms on Trees and Graphs*. Springer-Verlag, Berlin, 2002.

[68] L.M. Wadley and A.M. Pyle. The identification of novel RNA structural motifs using COMPADRES : an automated approach to structural discovery. *Nucleic Acids Research*, 32 :6650–6659, 2004.

[69] X. Wang, J. Huan, J.S. Snoeyink, and W. Wang. Mining RNA tertiary motifs with structure graphs. In *19th International Conference on Scientific and Statistical Database Management*, pages 31–31. IEEE, 2007.

[70] E. Westhof and P. Auffinger. RNA Tertiary structure. *Encyclopedia of Analytical Chemistry. R.A. Meyers (Ed)*, pages 5222–5232, 2000.

[71] E. Westhof and V. Fritsch. RNA folding : beyond Watson-Crick pairs . *Structure*, 8 :R55–R65, 2000.

[72] E. Westhof, B. Masquida, and L. Jaeger. RNA Tectonics : towards RNA design. *Folding and Design*, 1(4) :R78–88, 1996.

[73] H.S. Wilf. *Algorithms and Complexity*. A K Peters, Ltd, 2002.

[74] H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H.M. Berman, and E. Westhof. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Research*, 31 :3450–3460, 2003.