HAL open science

Algorithmes pour l' étude de la structure secondaire des ARN et l'alignement de séquences

Lou Feng

To cite this version:

Lou Feng. Algorithmes pour l' étude de la structure secondaire des ARN et l'alignement de séquences. Bio-informatique [q-bio.QM]. Université Paris Sud - Paris XI, 2012. Français. NNT : . tel-00781416

HAL Id: tel-00781416

https://theses.hal.science/tel-00781416v1

Submitted on 26 Jan 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.

N^o d'ordre :

THÈSE

présentée

devant l'Université Paris-Sud 11

pour obtenir

Le grade de : DOCTEUR EN SCIENCES DE L'UNIVERSITÉ PARIS-SUD 11

Mention INFORMATIQUE

Par

Feng LOU
Équipe d'accueil : Bioinformatique-LRI
École Doctorale : Informatique

Titre de la thèse :

Algorithmes pour l'étude de la structure secondaire des ARN et l'alignement de séquences

Soutenue le 30 Janvier 2012 devant la commission d'examen

M. : Pascal Ferraro ..... Rapporteur
M. : François Major ..... Rapporteur
M. : Abdel Lisser ..... Examinateur
M. : Jean-Marc Steyaert ..... Examinateur
M. : Peter Clote ..... Co-directeur De Thèse
M. : Alain Denise ..... Co-directeur De Thèse

Table des matières

1 Introduction **3**

2 Notions préliminaires **6**

2.1 Structures et fonctions de l'ARN 6

2.1.1 Structures d'ARN 6

2.1.2 Fonctions d'ARN 8

2.1.3 Prédiction de structures d'ARN 9

2.2 Distribution de Boltzmann 13

3 Prédiction de la thermodynamique des structures d'ARN par échantillonnage de Wang-Landau **14**

3.1 Introduction 14

3.2 Chaînes de Markov 15

3.2.1 Probabilité de transition et matrice de transition 16

3.2.2 Classification des états 16

3.2.3 Convergence en une distribution stationnaire 17

3.3 Méthode de Monte-Carlo 18

3.3.1 L'algorithme de Metropolis 20

3.3.2 La généralisation de Hastings 22

3.3.3 Recuit simulé 24

3.3.4 L'échantillonnage de Wang-Landau 25

3.4 Méthode de `RNA-WL` 28

3.4.1 Algorithme 29

3.4.2 Simulation de la chaîne de Markov 30

3.4.2.1 Densité d'états de l'énergie libre : $g(E_i)$ 31

3.4.2.2 Histogramme de l'énergie libre : $h(X_i)$ et Facteur de modification : f 32

3.4.3 Condition d'équilibre 33
3.4.4 Densité d'états de l'énergie 34
3.5 Résultats 37
3.5.1 Description du programme `RNA-WL` 37
3.5.2 Prédiction de la densité d'états d'énergie pour une molécule d'ARN . . . 38
3.5.2.1 Validation 38
3.5.2.2 Diversité structurale 41
3.5.2.3 Loi Normale ou distribution des valeurs extrêmes ? 43
3.5.2.4 Temps d'exécution. 44
3.5.3 Prédiction de la température de dénaturation pour l'hybridation de deux molécules d'ARN 47
3.5.3.1 Pipeline pour calculer la capacité de chaleur et la température de dénaturation 48
3.5.3.2 Prédiction de la température de dénaturation. 53
3.6 Discussion 54

4 Les structures MEA à différentes distances de paires de bases d'une structure secondaire d'ARN 55
4.1 Introduction 55
4.2 Riboswitch 56
4.3 Le programme `RNAbor` 59
4.4 La structure secondaire MEA 61
4.5 Méthode de `RNAborMEA` 62
4.5.1 Les structures $MEA(k)$ 63
4.5.2 Méthode 63
4.5.3 Algorithme 68
4.6 Résultats 71
4.6.1 Détection des structures fonctionnelles de Riboswitch 71
4.6.1.1 Riboswitch TPP 71
4.6.1.2 Comparaison entre les structures $MFE(k)$ et les structures $MEA(k)$ 73
4.6.1.3 Riboswitch Purine 74
4.6.2 Loi de pseudo-Boltzmann 80
4.7 Discussion 83

5 Une nouvelle méthode de recherche d'alignements deux-à-deux sous-optimaux **85**

5.1 Introduction 85

5.2 Alignement de séquences 86

5.2.1 Score d'alignement 88

5.2.1.1 Formule de score 88

5.2.1.2 Matrice de substitution 88

5.2.1.3 Pénalités de gap 90

5.2.2 Alignement global et local 91

5.2.3 Algorithme d'alignements sous-optimaux 93

5.3 Méthode de `SubOpt` 95

5.3.1 Distance entre deux alignements 95

5.3.2 Méthode de `SubOpt` 96

5.3.2.1 Entrées et Sorties 96

5.3.2.2 Matrice de score d'alignement 97

5.3.2.3 k-alignement 100

5.4 Résultats 101

5.4.1 *BAliBASE* 101

5.4.2 Comparaison avec l'alignement global 101

5.4.3 Comparaison des 3 méthodes d'alignement sous-optimal 104

5.4.3.1 Fréquence et entropie de position spécifique 105

5.4.3.2 Diversité sous-optimaux et corrélation avec la région fiablement alignée 108

5.5 Discussion 111

6 Conclusion **112**

7 Annexe **114**

Chapitre 1

Introduction

Tous les organismes vivants sont les produits d'un développement cellulaire. Grâce au patrimoine génétique, les cellules se développent et aquérir leurs nombreuses fonctionns. Trois types de macromolécules fondamentales sont impliquées dans ce processus : ADN, ARN et protéine. Dans le dogme central [1], un gène (ADN) est transcrit en ARN, qui lui même est traduit en protéine.

Dans cette conception, l'ARN n'a qu'un rôle d'intermédiaire. Mais en fait, il est apparu récemment que l'ARN est impliqué dans de nombreux processus biologiques dont les rôles étaient jusqu'alors insoupçonnés, comme par exemple dans la retraduction du code génétique [2], la régulation de la transcription et de la traduction des gènes [3, 4], la régulation de l'épissage alternatif [5], *etc.*

Pour comprendre les mécanismes d'action de l'ARN, il est important de connaître sa structure tridimensionnelle, car ils lui sont étroitement liés. Il existe différentes méthodes expérimentales permettant de découvrir la structure tertiaire d'ARN. Ces méthodes sont encore trop coûteuses en temps et en argent, la prédiction directe de la structure tertiaire est cependant un problème difficile. On peut plutôt étudier une autre représentation plus simplifiée des structures d'ARN, qui est la structure secondaire.

La prédiction de structures secondaires avec pseudonoeuds est un problème NP-complet [6]. En revanche, il existe des algorithmes de complexité polynomiale pour la prédiction de structures secondaires sans pseudonoeuds ou pour certaines classes de pseudonoeuds. On peut utiliser ces algorithmes pour calculer l'énergie libre minimum de structures d'une molécule d'ARN ou d'une hybridation de deux molécules d'ARN. Ces méthodes peuvent être groupées

en deux types d'algorithmes qui utilisent 1) soit une grammaire non contextuelle probabiliste pour calculer un modèle covariant, comme les programmes `Infernal` [7] et `Pfold` [8] 2) soit les paramètres expérimentals de l'énergie libre, comme les programmes `mfold` [9], `RNAfold` [10], `RNAstructure` [11], `UNAFold` [12] et `RNAcofold` [13, 14].

Le but général de ma thèse est de développer et implémenter des algorithmes pour avancer dans la prédiction de la structure de l'ARN d'une part, dans la comparaison de séquences d'autre part.

Un algorithme de Monte Carlo non-Boltzmann a été conçu par Wang et Landau pour estimer la densité d'états pour les systèmes complexes, tels que le modèle d'Ising. Dans le chapitre 3, nous appliquons la méthode Wang-Landau (WL) pour calculer la densité d'états des structures secondaires d'une séquence d'ARN donnée, et pour les hybridations de deux molécules d'ARN. Ce travail a été publié dans l'article :

Feng Lou, Peter Clote. Thermodynamics of RNA structures by Wang-Landau sampling. ISMB 2010, Bioinformatics 2010 Jun 15 ;26(12) :i278-86.

Dans le chapitre 4, nous étudions une classe d'ARN non-codant. On l'appelle riboswitch (riborégulateur), car il joue un rôle important dans un certain nombre de processus biologiques en effectuant un changement allostérique, c'est-à-dire une commutation entre deux structures distinctes. Il est donc important de développer un algorithme pour les prédire. Nous développons un nouvel algorithme de programmation dynamique qui engendre des structures sous-optimales, dédié principalement à la prédiction des deux structures fonctionnelles des riboswitchs. Ce travail a été publié dans les articles :

Feng Lou, Peter Clote. Maximum expected accurate structural neighbors of an RNA secondary structure. Proceedings of 1st IEEE International Conference on Computational Advances in Bio and medical Sciences (ICCABS), Feb 3-5, 2011 in Orlando, FL.

Peter Clote, Feng Lou, William A. Lorenz. Maximum expected accurate structural neighbors of an RNA secondary structure. (Version longue), accepté par BMC Bioinformatics.

L'alignement de séquences deux-à-deux [15, 16, 17, 18, 19] est un autre problème qui est étudié depuis plus de quarante ans. Il est utilisé dans beaucoup de problèmes en bioinformatique : identifier les sites fonctionnels, prédire la fonction d'une protéine, prédire la structure secondaire ou tertiaire d'une protéine ou d'un ARN, établir une phylogénie. En effet, pour un niveau d'identité de séquences de 10-15%, il reste un écart important entre la précision

de l'alignement de deux séquences et celle de l'alignement de deux structures tertiaires. Dans le chapitre 5, nous présentons un algorithme de recherche d'alignement sous-optimaux de séquences pour améliorer la qualité d'alignement de séquences. Ce travail a été publié dans l'article :

Peter Clote, Feng Lou, and Alain Denise. A new approach to suboptimal pairwise sequence alignment. IASTED conference CompBio 2011, July 11-13, 2011, Cambridge, UK.

Chapitre 2

Notions préliminaires

2.1 Structures et fonctions de l'ARN

L'ARN, ou acide ribonucléique, est une macromolécule monocaténaire, composée de l'assemblage d'une succession de nucléotides. Ces nucléotides sont constitués d'un acide phosphorique, d'un sucre (le ribose) et d'une base azotée parmi l'adénine, la cytosine, la guanine et l'uracile, communément notées A, C, G et U, respectivement.

Les bases azotées ont la faculté de s'apparier entre elles par des liaisons hydrogène. On observe fréquemment les appariements A-U, U-A, C-G et G-C, dits appariements canoniques ou Watson-Crick, et les appariements G-U et U-G, dits appariements wobble. Beaucoup d'autre appariements sont observés dans les structures d'ARN, ils sont appelés des appariements non-canoniques [20, 21].

2.1.1 Structures d'ARN

Cette possibilité d'appariements confère à la molécule d'ARN la capacité de se replier et de prendre une conformation spatiale complexe, que l'on appelle sa structure. On distingue plusieurs niveaux de complexité dans le repliement d'un ARN.

La structure primaire d'ARN est la séquence de nucléotides du début à la fin de la molécule, qui est composée par un mot de l'alphabet {A, C, G, U}.

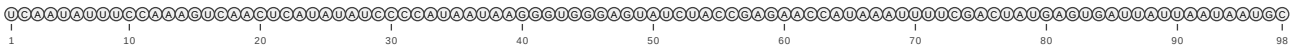


FIGURE 2.1 – Structure primaire d'un ARN de la famille riboswitch purine, dont l'identifiant EMBL est CP001581.1_3749039-3748942.

La structure secondaire d'ARN peut être définie par un sous-ensemble de liaisons hydrogène pouvant être dessinées dans le plan sans croisement.

Un triplet de bases dans s correspond à deux paires de bases $(a_i, a_j), (a_i, a_l) \in s$ ou $(a_i, a_j), (a_k, a_j) \in s$, un pseudonoeud dans s correspond deux paires de bases $(a_i, a_j), (a_k, a_l) \in s$, telque $i < k < j < l$ ou $k < i < l < j$.

Le pseudonoeud est un motif de repliement comprenant des bases libres et des paires de bases. Il met en jeu au moins deux hélices et se caractérise par l'apparition d'un croisement dans la représentation linéaire des structures secondaires.

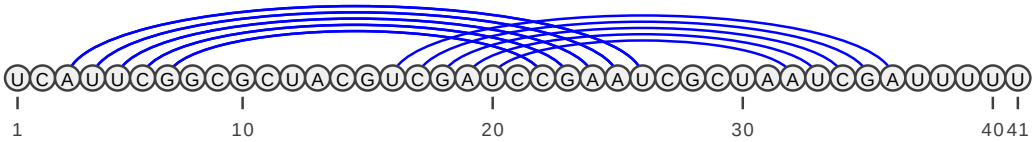


FIGURE 2.2 – Structure secondaire avec pseudonœud d'un ARN en représentation linéaire.

Définition 2.1 *Soit $a = \{a_1, \ldots, a_n\}$ $(\forall 1 \leq i \leq n, a_i \in \{A, C, G, U\})$ une séquence (structure primaire) d'ARN. Une structure secondaire s de a est un ensemble de paire de bases (a_i, a_j), tel que a_i et a_j forment une paire de bases Watson-Crick ou Wobble, et tel qu'il n'y a ni du triplet de bases ni du pseudo-noeud dans s*

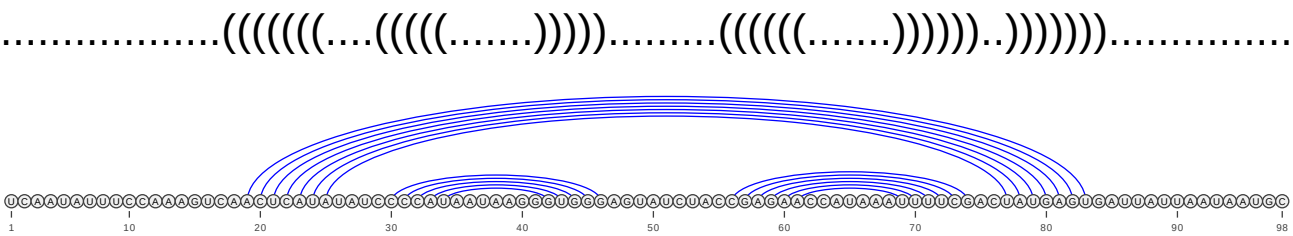


FIGURE 2.3 – Structure secondaire d'un ARN : représentation point-parenthèse (en haut), représentation linéaire (en bas).

La structure tertiaire d'un ARN est finalement sa forme dans l'espace, qui correspond à l'ensemble des liaisons (appariements canoniques et non-canoniques et d'autre types de liaisons comme l'empilement) intervenant dans le repliement d'un ARN et décrit sa structure tridimensionnelle. Dans la figure 2.4, nous voyons une chaîne de nucléotides avec des bases azotés différents, où les sphère gris sont des atomes de carbone, les sphère bleu sont des atomes d'azote, les sphère rouge sont des atomes d'oxygène, les sphère orange sont des atomes de phosphore. A ce niveau de structures peuvent figurer des liaisons triples ou quadruples, autrement dit, un nucléotide peut être apparié à plus d'un seul autre nucléotide.

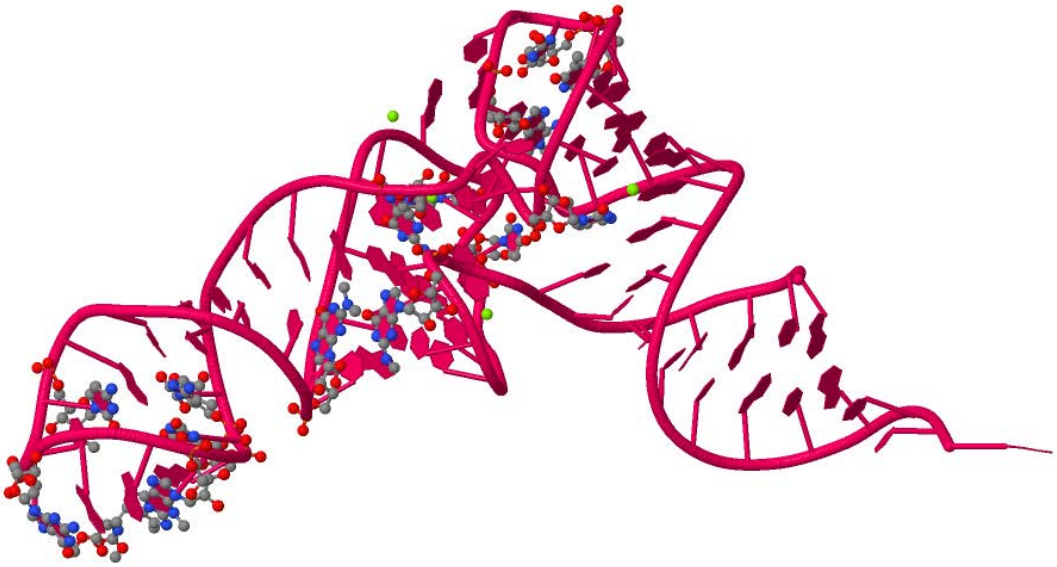


FIGURE 2.4 – Structure tertiaire dARNt.

Le dernier niveau de complexité est la structure quaternaire, qui décrit les interactions entre ARN et éventuellement protéines au sein d'un complexe.

2.1.2 Fonctions d'ARN

Les ARN commencent tout juste être considés comme un acteur de la synthèse des protéines. Lorsque les gènes de l'ADN sont transcrits en ARN, certains d'entre eux, dits ARN messagers, sont ensuite traduits en protéines. Les ARN de transfert et les ribosomes, complexes composés en partie d'ARN ribosomiques, entrent en jeu lors de la traduction des ARN messagers en

protéines.

Cependant, la grande révolution de l'ARN vient de la découverte des ARN non codants (ARNnc). Ceux-ci représentent en fait la majorité des ARN transcrits par la cellule. Ces ARN ne sont pas traduits en protéines et interviennent directement dans plusieurs voies métaboliques. Parmi les nombreuse familles d'ARNnc, on trouve par exemple les suivantes :

- Les micro-ARN, une catégorie d'ARNnc, sont des ARN simple-brin, longs d'environ 21 24 nuclotides, ils jouent un rôle important dans la régulation de l'expression des gènes. Les micro-ARN régulent négativement la traduction en s'appariant avec des ARNm. On pense actuellement que 30% de l'ADN codant est ainsi régulé par des micro-ARN [22]. Leur importance s'est avérée dans de nombreux processus comme la croissance [23], la différentiation cellulaire [24], l'apoptose [25], et la prolifération cellulaire [26].
- Les petits ARN interférent (pARNi), une catégorie d'ARNnc, sont des petits ARN de 2025nt contenant les deux brins qui sont produits par clivage d'une ribonucléase de type III appelée Dicer (l'éminceuse). Dicer transfère alors les petits ARN interférents à un gros complexe multiprotéique, le complexe RISC (RNA-induced silencing complex). Le complexe RISC clive l'ARNm cible qui va être alors dégradé et n'est donc plus traduit en protéine. Ce mécanisme est trés spécifique de la séquence du siRNA et de sa cible, l'ARNm.
- Les riboswitchs (les riborégulateurs) constituent une autre catégorie d'ARNnc qui démontre les capacités catalytiques de l'ARN (voir la section 4.2). Les riboswitchs sont des ARN capables de changer de conformation en réponse à certains stimuli de manière à exercer une fonction de régulation dans la cellule. Ces stimuli peuvent être par exemple la température [27] ou l'interaction avec un métabolite [84].

2.1.3 Prédiction de structures d'ARN

Pour comprendre les mécanismes d'action de l'ARN, il est important de connaître sa structure tridimensionnelle, car ils sont étroitement liés. Il existe différentes méthodes expérimentales permettant de découvrir la structure tertiaire d'ARN. Ces méthodes sont encore trop coûteuses en temps et en argent, la prédiction directe de la structure tertiaire est cependant un problème difficile bien que quelques travaux ont fait des avancées notables [29]. Une étape intermédiaire prometteuse est celle de la prédiction de structures secondaires. La connaissance de la structure secondaire apporte en effet une information très précieuse sur la structure tertiaire de l'ARN. Il

a été montré expérimentalement que le repliement des ARN suit un processus hiérarchique : la structure primaire code la structure secondaire qui elle-même sert de support à la structure tertiaire [30]. La structure secondaire apporte la principale contribution énergétique qui explique la structure tertiaire.

Plusieurs approches ont été proposées pour la prédiction de structure secondaire d'ARN, elles sont basées soit sur des méthodes comparatives, soit sur des modèles d'énergie.

- L'approche comparative s'applique à un ensemble de séquences d'ARN homologues ou d'un alignement des séquences d'ARN, et est basée sur le fait que la structure secondaire est plus préservée que la séquence [31, 32, 33]. Elle recherche une conformation de score (énergie + alignement) élevé.
- L'approche thermodynamique consiste à trouver, à partir d'une séquence d'ARN et d'un modèle énergétique, le repliement le plus stable en calculant son énergie libre minimum.

Dans l'approche thermodynamique, la méthode la plus simple est l'algorithme de programmation dynamique de Nussinov [34], qui consiste à replier une séquence d'ARN en maximisant le nombre d'appariements et qui adopte une fonction d'énergie E qui ne prend en compte que les appariements, pas l'empilement. Pour chaque sous-séquence allant de la position i à j, on calcule l'énergie $E(i,j)$ de la structure la plus stable $s(i,j)$ qui peut être obtenue par les 4 façons suivantes :

1. ajouter i, j, appariés, à $s(i+1, j-1)$.
2. ajouter i, non-apparié, à $s(i+1, j)$.
3. ajouter j, non-apparié, à $s(i, j-1)$.
4. réunir deux sous-structures optimales, $s(i,k)$ et $s(k+1,j)$.

L'énergie $E(i,j)$ peut donc calculée par la récurrence suivante :

$$E(i,j) = \min \begin{cases} E(i-1,j-1) + e(i,j) \\ E(i+1,j) \\ E(i,j-1) \\ \min\limits_{i<k<j} E(i,k) + E(k+1,j). \end{cases} \tag{2.1}$$

où $e(i,j)$ est l'énergie de la paire de bases i et j.

Cependant, cette méthode repose sur un modèle d'énergie qui n'est pas réaliste. Dans la réalité des structures d'ARN, l'énergie libre ne dépend pas seulement des paires de bases, mais

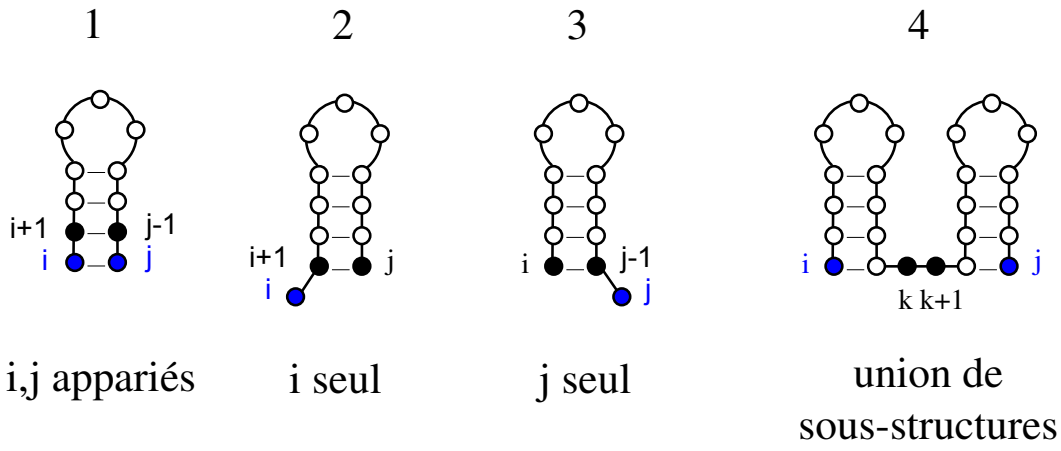


FIGURE 2.5 – Les 4 façons pour obtenir la structure $S(i, j)$.

aussi des empilements, des boucles internes, des épingles à cheveux, des renflements, et des boucles multiples (voir la figure 2.6).

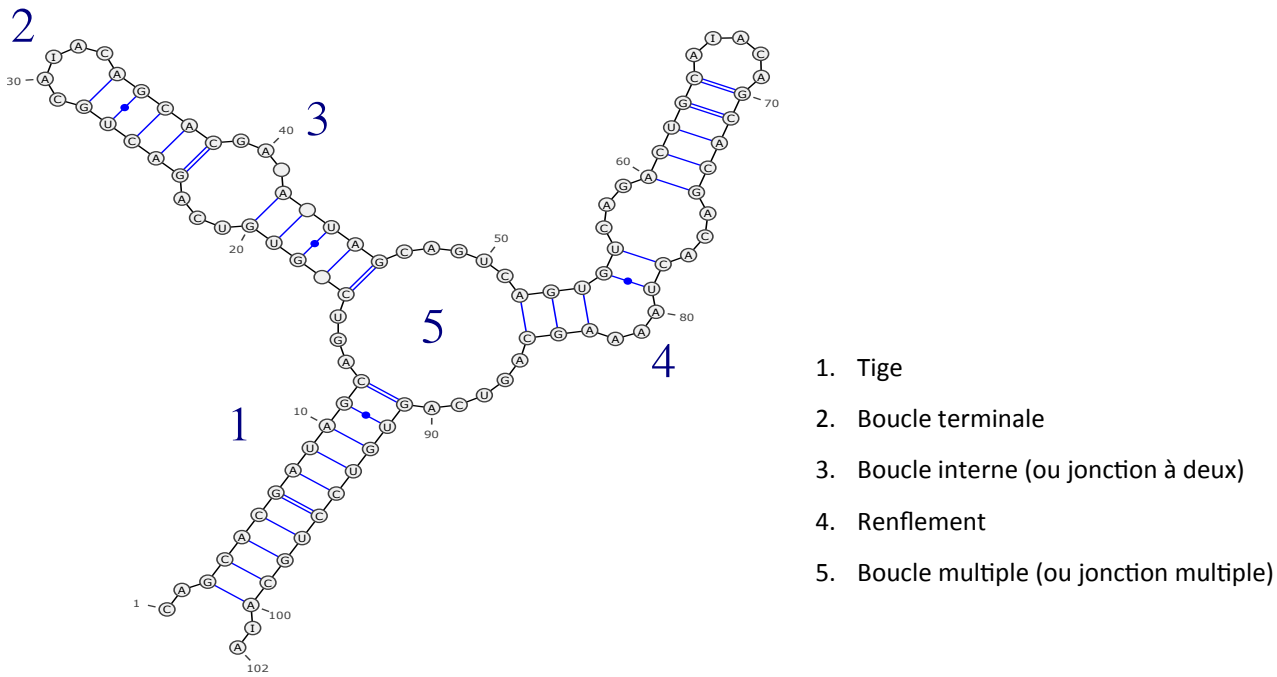


FIGURE 2.6 – Motifs structuraux d'ARN.

En 1981, Zuker *et al.* [9] ont proposé un autre modèle plus réaliste en prenant compte tous ces motifs structuraux d'ARN. Par rapport au modèle de Nussinov-Jacobson, le modèle de Zuker va prendre deux matrices au lieu d'une pour calculer l'énergie de structure.

Soit $E(i, j)$ l'énergie de la meilleure structure sur la sous-séquence entre les positions i et j.

$$E(i,j) = \min \begin{cases} E(i+1,j) \\ E(i,j-1) \\ V(i,j) \\ \min\limits_{i<k<j} E(i,k) + E(k+1,j). \end{cases} \tag{2.2}$$

Soit $V(i,j)$ l'énergie de la meilleure structure sur la sous-séquence entre les positions i et j où l'appariement (i,j) est formé.

$$V(i,j) = \min \begin{cases} \textit{Épingle à cheveux}(i,j) \\ V(i+1,j-1) + e(i,j) \\ \textit{Boucle interne}(i,j) \\ \textit{Boucle multiple}(i,j). \end{cases} \tag{2.3}$$

Dans cette équation, $e(i,j)$ est l'énergie de l'empilement $(i, i{+}1, j{-}1, j)$, *Épingle à cheveux*(i,j) [resp. *Boucle interne*(i,j), *Boucle multiple*(i,j)] est l'énergie de la structure où les bases i et j sont appariées, la paire de bases est située dans un épingle à cheveux [resp. boucle interne, boucle multiple].

Les données expérimentales pour les énergies des motifs structuraux s'accumulant petit à petit au cours des années, le modèle a été paramétré progressivement. On l'appelle le modèle de Turner. Les valeurs les plus utilisées aujourd'hui ont été établies dans les articles [35, 36], et elles ont été révisées dans l'autre article [11]. Cet algorithme est ensuite implémenté dans les programmes, comme par exemple : Mfold [37] et RNAfold [38].

D'autre part, le problème de la prédiction de structures secondaires ayant pseudonoeuds de certaines classes, comme PKF, L&P, D&P, *etc.*, a été résolu en temps polynomial [39].

Plus récemment, Major *et al.* ont construit un pipeline MC-Fold/MC-sym [29], qui prédit la structure secondaire et aussi la structure tertiaire à partir d'une séquence d'ARN. Dans la prédiction de la structure secondaire, il recherche des structures qui contiennent non seulement des paires de bases canoniques (A-U, G-C, G-U), mais aussi les autres paires de bases non-canoniques. L'énergie de la structure secondaire est calculée en accumulant les énergies de tous les "cycles", qui sont obtenues par les statistiques de l'ensembles des structures tertiaires dans la base de données PDB (Protein Date Bank).

2.2 Distribution de Boltzmann

Dans la dernière partie du 19e siècle, L.Boltzmann [1] a démontré que dans les conditions d'une gaz parfait avec N molécules, le nombre N_i de molécules ayant l'énergie E_i satisfait :

$$N_i = N * \frac{e^{\frac{-E_i}{kT}}}{Z} \tag{2.4}$$

où k est la constante des gaz parfaits ($1.986 \cdot 10^{-3}\ kCal \cdot Kelvin^{-1} \cdot mol^{-1}$), T est la température absolue en Kelvin, Z est la fonction de partition qui satisfait :

$$Z = \sum_{i \in \Omega} e^{\frac{-E_i}{kT}} \tag{2.5}$$

où Ω est l'ensemble des états d'énergie.

Plus récemment, la distribution de Bolzmann a été utilisée pour pondérer les structures d'une séquence d'ARN par un facteur de Boltzmann : $e^{-E(s)/k \cdot T}$.

La probabilité de Boltzmann $p_{s,w}$ d'une structure s pour une séquence w est alors donnée par :

$$p_{s,w} = \frac{e^{-E(s)/k \cdot T}}{Z_w} \tag{2.6}$$

où Z_w est la fonction de partition des énergies de l'ensemble des structures compatibles avec w, c'est la somme des facteurs de Boltzmann de toutes les structures de w.

Dans [40], Ding *et al.* prédisent la structure d'ARN en faisant les trois étapes suivantes : 1) échantillonner des structures selon une probabilité de Boltzmann. 2) effectuer un clustering. 3) construire la structure consensus dans le plus grand cluster. Par rapport à la prédiction en minimisant l'énergie libre, ils observent une amélioration relative pour la spécificité (+17.6%) et la sensibilité (+21.74%, sauf Introns du groupe II).

Soit S_w l'ensemble de structures secondaire de w, Ω l'ensemble des états d'énergie de S_w.

Définition 2.2 ***La densité d'états d'énergie*** *d'une séquence d'ARN w, notée par $g(E_i)$, est le nombre de structures ayant leur énergie dans l'intervalle E_i, divisé par le nombre de structures de w.*

1. Ludwig Boltzmann, physicien autrichien, 1844-1906

Chapitre 3

Prédiction de la thermodynamique des structures d'ARN par échantillonnage de Wang-Landau

3.1 Introduction

Comme nous avons vu au chapitre 2, les algorithmes de programmation dynamique basés sur les paramètres thermodynamiques des structures secondaires d'ARN ont une immense importance dans la biologie moléculaire. Ils ont été développés pour calculer l'énergie libre minimume de la structure secondaire et la fonction de partition d'une séquence d'ARN ou d'une hybridation des deux molécules d'ARN. Cependant, l'applicabilité des méthodes de programmation dynamique nécessite d'interdire certains types d'interactions, comme certaines classes de pseudonoeuds, les zig-zags [41], etc.

Dans ce chapitre, nous nous consacrons à un nouvel algorithme `RNA-WL`, qui est une application de l'algorithme de Wang Landau pour la prédiction de la densité d'états d'énergie des structures secondaires d'une molécule ou d'une hybridation de deux molécules d'ARN.

Avant de discuter comment fonctionne notre algorithme `RNA-WL`, nous allons d'abord présenter les principes de la chaîne de Markov et de certaines méthodes de Monte Carlo, y compris l'algorithme de Metropolis, la généralisation de Hastings, le recuit simulé et l'algo-

rithme original de Wang Landau. Ensuite, nous allons montrer les résultats en deux parties. La première partie est pour la recherche de la densité d'états d'énergie de toutes les structures secondaires d'une molécule d'ARN donnée. La densité obtenue par notre programme `RNA-WL` est validée par la densité exacte obtenue par le programme `RNAsubopt`. Nous allons aussi montrer que la performance de notre programme `RNA-WL` est meilleure par rapport aux autres programmes existants en temps d'exécution et en la taille maximale de séquence d'ARN qui peut être traitée. La deuxième partie est pour estimer la température de dénaturation pour l'hybridation de deux molécules d'ARN. Nous allons construire un pipeline qui utilise la densité d'états obtenues par notre algorithme `RNA-WL` et calcule la température de dénaturation. À la fin, nous allons montrer que la plupart des températures de dénaturation obtenues par notre programme `RNA-WL` sont plus proches des valeurs expérimentales que les autres programmes existants.

3.2 Chaînes de Markov

Définition 3.1 *Une chaîne de Markov à temps discret $M = (\Omega, \pi, P)$ est un processus stochastique $\{X_n, n = 0, 1, \ldots\}$ à temps discret, défini sur un espace d'états Ω fini ou dénombrable, ayant une distribution initiale π et vérifiant la propiété de Markov :*

$$P[X_n = i | X_0, \ldots, X_{n-1}] = P[X_n = i | X_{n-1}] \tag{3.1}$$

pour tout $i \in \Omega$ et quel soit $n \geq 1$.

En mots, l'état courant résume, à lui seul, tout l'historique du système susceptible d'influencer son évolution future.

Définition 3.2 *Une chaîne de Markov à temps discret est* ***homogène*** *(dans le temps) si, pour toute paire d'états (i, j) et pour tous n, k, où $n \geq \max(1, 1 - k)$,*

$$P[X_n = j | X_{n-1} = i] = P[X_{n+k} = j | X_{n+k-1} = i]. \tag{3.2}$$

3.2.1 Probabilité de transition et matrice de transition

Pour une chaîne de Markov homogène $\{Xn, n = 0, 1, \ldots\}$, on a

$$P[X_n = j|X_{n-1} = i] = P[X_1 = j|X_0 = i] \ \ \forall n \geq 1. \tag{3.3}$$

Définition 3.3 *La probabilité de transition (en 1 étape) de i à j est définie comme suit :*

$$p_{ij} = P[X_1 = j|X_0 = i] \ \ \forall i, j \in \Omega. \tag{3.4}$$

En mots, la probabilité p_{ij} est égale à la probabilité conditionnelle que le système se retrouve dans l'état j à l'étape suivante, sachant qu'il se trouve actuellement dans l'état i. Si la chaîne possède $n = |\Omega|$ états, les probabilités précédentes peuvent être rangées dans une matrice de transition $P = (p_{ij})$ de taille $n \times n$ dont les lignes et les colonnes sont indexées par les éléments de Ω.

Une matrice carrée $P = (p_{ij})$ est stochastique si :

- ses éléments sont non négatifs : $p_{ij} \geq 0$ pour tout i et j.
- la somme des éléments de chacune de ses lignes est égale à 1 : $\sum_j p_{ij} = 1$, pour tout i.

Définition 3.4 *La probabilité conditionnelle d'aller de i à j en m étapes exactement est*

$$p_{ij}^{(m)} = P[X_m = j|X_0 = i] = P[X_{n+m} = j|X_n = i] \ \ \forall n \geq 1. \tag{3.5}$$

Cette probabilité est indépendante de n car le processus est homogène et est appelée la probabilité de transition en m étapes de i à j. La matrice $P^{(m)}$ dont l'élément (i, j) est égal à $p_{ij}^{(m)}$ est appelée la matrice de transition en m étapes. On note $P^{(m)} = P^m$, la m-ième puissance de P.

3.2.2 Classification des états

Définition 3.5 *Soient i et j deux états de Ω. L'état j est* ***accessible*** *à partir de i si et seulement si :*

$$\exists n \geq 0, P^n(i, j) = P(X_n = j|X_0 = i) > 0. \tag{3.6}$$

On dit que les états i et j **communiquent** et on note $i \Leftrightarrow j$ si et seulement si j est accessible à partir de i et i est accessible à partir de j.

La relation $i \Leftrightarrow j$ est une relation d'équivalence sur Ω. L'espace Ω peut donc être partitionné en classes d'équivalence pour la relation $i \Leftrightarrow j$, appelées **classes d'états communicants**.

Définition 3.6 *La chaîne est **irréductible** si chaque état est accessible à partir de chaque autre état, ou autrement dit lorsque l'espace d'états Ω est réduit à une seule classe (cas où tous les états communiquent entre eux).*

$$\forall i, j \in \Omega, \ \ \exists N \geq 0, \ \ p_{i,j}^{N} > 0. \tag{3.7}$$

Définition 3.7 *L'état i de Ω est **récurrent** si et seulement si, partant de i , la chaîne X revient presque sûrement à l'état i.*

$$\sum_{n=0}^{\infty} p_{i,i}^{(n)} = \infty. \tag{3.8}$$

Un état non récurrent est transient.

Définition 3.8 *La **période** d'un état i est l'entier*

$$d(i) = PGCD\{n \geq 1 | P_{(i,i)}^{n} > 0\} \tag{3.9}$$

*La chaîne de Markov M est **apériodique** si et seulement si tous ses états sont de période 1.*

3.2.3 Convergence en une distribution stationnaire

Théorème 3.1 *Si la chaîne de Markov $M = (\Omega, \pi, P)$ est finie, apériodique, irréductible, où Ω contient n états, alors, il existe une distribution stationnaire :*

$$\lim_{t \to \infty} p_i^{(t)} = p_i^* \tag{3.10}$$

où p_i^* est l'unique solution qui est pour les conditions :

- $p_i^* \geq 0$
- $\sum_{i=1}^n p_i^* = 1$
- $p_j^* = \sum_{i=1}^n p_i^* p_{i,j} = 1$

La distribution $(p_1^*, \ldots, p_n^*)$ *est aussi appelée* ***la distribution stationnaire****.*

Supposons que la chaîne de Markov $M = (\Omega, \pi, P)$ a une distribution stationnaire $(p_1^*, \ldots, p_n^*)$. M est **réversible** si

$$\forall (i,j) \in \Omega, \quad p_i^* p_{i,j} = p_j^* p_{j,i} \tag{3.11}$$

L'équation 3.11 est aussi appelée la condition d'équilibre. [1]

Comme nous le verrons par la suite, l'existence d'une distribution stationnaire n'est pas seulement un théorème mathématiquement intéressant, mais il fournit plutôt la justification de la convergence de l'algorithme de "Markov Chaine Monte Carlo" (MCMC), où un échantillonnage est effectué sur une chaîne de Markov apériodique et irréductible $M = (\Omega, \pi, P)$, dont les probabilités de la distribution limite stationnaire sont données par la distribution de Boltzmann : $p_i^* = \frac{e^{-E(i)/RT}}{Z}$, où $E(i)$ est l'énergie de l'état i, T est la température absolue en Kelvin, R est la constante universelle des gaz parfaits, et Z est la fonction de partition tel que $Z = \sum_{i \in \Omega} e^{E(i)/RT}$.

3.3 Méthode de Monte-Carlo

La méthode de Monte-Carlo désigne toute méthode visant à calculer une valeur numérique en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. Cette méthode a été initialement développée par Metropolis *et al.* [42, 43], pour calculer les propriétés d'équilibre des systèmes physiques [44, 45, 46].

La méthode de Monte-Carlo a un rôle important dans l'évaluation des intégrales et la simulation de systèmes stochastiques. Par exemple, nous pouvons déterminer la valeur de π (pi) comme suit :

On engendre n points M_i de coordonnées (x_i, y_i), où $0 < x_i < 1$ et $0 < y_i < 1$. Soit Z le

1. "detailed balance condition" en anglais

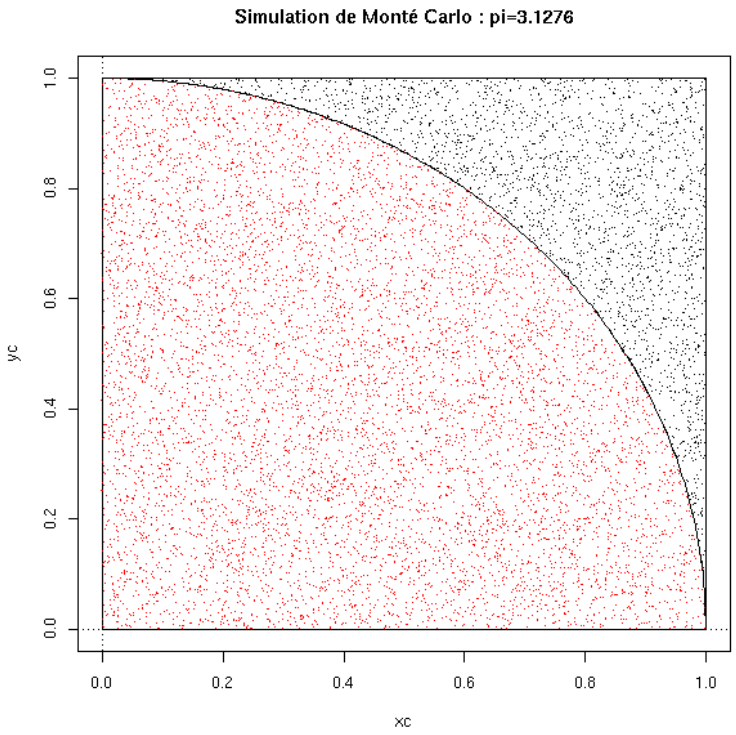


FIGURE 3.1 – L'illustration du tirage aléatoire pour calculer la valeur de π, cette image est extraite du site $http : //zoonek2.free.fr/UNIX/48_R_2004/14.html$

nombre des points observés dans le cercle de rayon 1. Pour $i \in 1, \ldots, n$, on définit la valeur aléatoire X_i à 1, si le point est dans le cercle, et à 0 sinon. Alors, $E(X) = \pi$, et l'estimateur Z/n est la même que la moyenne empirique de X_i qui est définie comme :

$$\bar{X_n} = \frac{X_1 + \ldots + X_n}{n} \tag{3.12}$$

Par le théorème centrale limite, un intervalle de confiance approximatif de 95% pour la valeur de π est :

$$[\bar{X_n} - 1.96\frac{\sigma}{\sqrt{n}},\ \bar{X_n} + 1.96\frac{\sigma}{\sqrt{n}}] \tag{3.13}$$

où σ est l'écart-type de X_i. Plus le nombre des points générés n est grand, plus la largeur de l'intervalle de confiance est petite, plus la valeur empirique $\bar{X_i}$ approche de la valeur de π.

L'étape la plus critique dans le développement d'un algorithme efficace du type Monte-Carlo est le tirage aléatoire à partir de la distribution de probabilité appropriées $\pi(x)$. Dans l'exemple précédent, nous connaissons la distribution $\pi(x)$ pour le tirage aléatoire. Quand la génération directe indépendante des échantillons à partir de $\pi(x)$ est impossible, comme dans le problème

de calculer la moyenne de Boltzmann, alors nous pouvons utiliser l'algorithme de Metropolis.

3.3.1 L'algorithme de Metropolis

En 1953, Nicholas Metropolis *et al* [44] ont proposé une nouvelle procédure d'échantillonnage qui incorpore une température du système. L'idée fondamentale de cet algorithme est de simuler l'évolution d'un processus de Markov pour atteindre l'échantillonnage de $\pi(x)$. Cette idée a été, plus tard, connue sous le nom d'algorithme de Metropolis. La première version de l'algorithme de Metropolis considérait le cas particulier de la distribution de Boltzmann, une des distributions les plus utilisées en physique statistique. En 1970, W. Keith Hastings [49] a étendu l'algorithme au cas de n'importe quelle distribution. Cet algorithme possède une grande simplicité et une grande puissance - des variations et extensions ont été largement adoptées par les chercheurs dans de nombreux domaines scientifiques, y compris la biologie, la chimie, l'informatique, l'économie, l'ingénierie, la science des matériaux, la physique, la statistique, *etc.*

L'algorithme de Metropolis est particulièrement important en physique statistique, où les systèmes ont un grand nombre de degrés de liberté et où les quantités d'intérêt, comme les moyennes thermiques, ne peuvent pas être calculées exactement. Dans un système avec d degrés de liberté, par exemple, la moyenne thermique d'une quantité A associée à chaque micro-état du système à l'équilibre à la température absolue T est donnée par :

$$\langle A \rangle = \frac{1}{Z} \int A(X) e^{-E(X)/RT} dx \tag{3.14}$$

où X est un point dans l'espace en D-dimensions, repésentant un état du système, $E(X)$ est l'énergie de l'état X, $Z = \int e^{-E(X)/RT} dx$ est la fonction de partition, R est la constante de Boltzmann. Dans le cas des réseaux de conformations de protéine [2], où l'espace conformationnel est discret, l'intégrale de l'équation ci-dessus est remplacée par une somme sur toutes les conformations :

$$\langle A \rangle = \frac{1}{Z} \sum_x A(X) e^{-E(X)/RT} \tag{3.15}$$

où les différents états du système X correspondent aux différentes conformations et la fonction de partition est : $Z = \sum_X e^{-E(X)/RT}$.

2. "protein lattice model" en anglais.

Dans le cas d'un système de petite taille, toutes les conformations peuvent être énumérées, et les moyennes thermiques (même les quantités extensives comme l'entropie et l'énergie libre) peuvent être calculées exactement par l'équation 3.15. Toutefois, si on essaye de traiter un système de grande taille, l'énumération complète de l'espace conformationnel est impossible. Avec la simulation de Monte Carlo Metropolis, cette difficulté est résolue par le remplacement de l'ensemble des conformations dans l'équation 3.15 par un sous-ensemble représentatif de conformations dont le nombre de conformations M est beaucoup plus petit que le nombre total de conformations N. Une estimation de la moyenne thermique $\langle A \rangle_{est}$ est obtenue par

$$\langle A \rangle_{est} = \frac{\sum_{i=1}^{M} A(X_i) \cdot e^{-E(X_i)/RT}}{\sum_{i=1}^{M} e^{-E(X_i)/RT}} \tag{3.16}$$

Clairement, la précision de l'estimation dépendra directement de la qualité du sous-ensemble représentatif de conformations.

Quand la probabilité d'occurrence d'une conformation donnée est proportionnelle à son facteur de Boltzmann, des échantillons représentatifs de conformations sont donc générés par l'algorithme de Metropolis.

L'algorithme construit une chaîne de Markov de conformations, où la première conformation X_0, est arbitrairement choisie (par exemple, au hasard) et une fonction de probabilité appropriées, appelée la loi de proposition ou la loi instrumentale $Q(X_{i-1} \rightarrow X_i)$, est utilisée pour construire chaque conformation X_i, à partir de la conformation précédente X_{i-1}. $Q(X_{i-1} \rightarrow X_i)$ est la probabilité d'un mouvement de la conformation X_{i-1} à la conformation X_i. Cette loi de proposition doit être symétrique, elle peut être par exemple une loi uniforme :

$$\forall (x_i, x_j) \in \Omega^2, Q(x_i, x_j) = \frac{1}{|\Omega| - 1} \tag{3.17}$$

où $|\Omega|$ est le nombre d'états dans Ω.

En général, pour faire une telle chaîne de conformations qui converge vers la distribution souhaitée, il suffit (mais ce n'est pas nécessaire) d'imposer la condition d'équilibre (voir. l'équation 3.11), selon laquelle l'égalité de la condition d'équilibre doit vérifier pour toute paire arbitraire de conformations, X_i et X_j.

La condition d'équilibre local donnée par l'équation 3.11 implique que, à l'équilibre, le nombre moyen de déplacements $X_i \rightarrow X_j$ est le même que le nombre moyen de déplacements inverses $X_j \rightarrow X_i$. Comme cela est vrai pour toute paire de conformations arbitraires, il s'ensuit que si le système n'est pas en équilibre alors le rapport entre les probabilités de deux confor-

mations tend à augmenter si elle est en dessous de sa valeur d'équilibre et de diminuer si elle est en dessus de sa valeur d'équilibre. Il s'ensuit que pour les simulations suffisamment longue le système va atteindre l'équilibre thermodynamique qu'on attend.

L'algorithme de Metropolis est comme suit. La première conformation est générée aléatoirement. les autres conformations vont être générées par l'itération des deux étapes suivantes :

1. Proposer aléatoirement une nouvelle conformation $X_{i'}$ à partir de la conformation actuelle X_i avec la probabilité uniforme $Q(X_i \to X_{i'}) = \frac{1}{N-1}$.
2. Accepter de passer à la conformation $X_{i'}$ avec la probabilité $r(X_i, X_{i'}) = min\{1, e^{-\Delta E/RT}\}$, où $\Delta E = E(X_{i'}) - E(X_i)$ est la différence entre l'énergie de la conformation proposée et l'énergie de la conformation actuelle.

   En pratique, nous faisons comme suit.

   On génère un nombre aléatoire $u \sim$ Uniforme[0, 1]. Nous décidons de passer à la conformation suivante $X_{i'}$ ou de rester à la conformation actuelle X_i selon la condition suivante :

$$X_{i+1} = \begin{cases} X_{i'} & \text{si} \quad u \leq \frac{\pi(X_{i'})}{\pi(X_i)} \\ X_i & \text{sinon} \end{cases}$$

   La probabilité de transition de l'état X_i à l'état $X_{i'}$ est : $p_(i, i') = Q(X_i \to X_{i'}) \cdot r(X_i, X_{i'})$.

L'algorithme de Metropolis est en fait une marche aléatoire sur l'espace des conformations. Les probabilités de transitions utilisées vont permettre d'atteindre la convergence vers la distribution stationnaire souhaitée et d'obtenir une estimation de la quantité d'intérêt.

Dans l'article de l'algorithme de Metropolis [44], les auteurs ont limité leur choix de la loi de proposition Q : la possibilité de choisir $X_{i'}$ à partir de X_i est toujours égale à celle de choisir X_i à partir de $X_{i'}$. Formellement, cette exigence de symétrie peut être exprimée comme

$$Q(X_i \to X_{i'}) = Q(X_{i'} \to X_i) \tag{3.18}$$

3.3.2 La généralisation de Hastings

L'algorithme de Metropolis simule une chaîne de Markov. Il utilise une loi de proposition Q symétrique pour suggérer un mouvement possible et emploie ensuite une règle de rejet.

Hastings [49] a étendu plus tard l'algorithme au cas où cette loi de proposition Q n'est pas nécessairement symétrique, la valeur de $Q(X_i \to X_{i'})$ peut être différente de la valeur de $Q(X_{i'} \to X_i)$.

Dans la généralisation de Hastings, la chaîne de Markov doit respecter les deux conditions suivantes : 1) La chaîne est irréductible, c'est-à-dire que tout état est atteignable depuis tout autre état. Cette condition assure qu'il y a au plus une distribution stationnaire asymptotique 2) La condition d'équilibre local, $p_i^* \cdot p_{i,i'} = p_{i'}^* \cdot p_{i',i}$, cette deuxième condition assure qu'il existe au moins une distribution asymptotique.

L'algorithme de Metropolis-Hastings va itérer les étapes suivantes :

- Choisir un état initial $X_0 \in \Omega$.
- Répéter M fois :
  - Proposer un autre état $X_{i'}$ selon la probabilité $Q(X_i \to X_{i'})$.
  - Générer un nombre $u \sim$ Uniform [0, 1], mettre à jour l'état.

$$X_{i+1} = \begin{cases} X_{i'} & \text{si} \quad u \leq r(x_i, X_{i'}) \\ X_i & \text{sinon} \end{cases}$$

où :

$$r(X_i, X_{i'}) = \min\{1, e^{-\Delta E/RT} \cdot \frac{Q(X_i \to X_{i'})}{Q(X_{i'} \to X_i)}\} \tag{3.19}$$

En fait, $\Delta E = E(X_{i'}) - E(X_i)$, $e^{-\Delta E/RT} = \frac{e^{-E(X_{i'})/RT}/Z}{e^{-E(X_i)/RT}/Z} = \frac{p^*_{X_{i'}}}{p^*_{X_i}}$. Ainsi, dans l'algorithme de Metropolis, on n'a pas besoin de connaître la fonction de partion de la distribution stationnaire.

Voici quelques remarques sur cet algorithme : 1) Si la condition $u \leq r(X_i, X_{i'})$ est rejetée, l'état actuel deviendra l'état suivant dans la chaîne de Markov : $X_{i+1} = X_i$. 2) Le calcul de la valeur de $r(X_i, X_{i'})$ ne dépend pas la fonction de partition de p_i^*, qui est souvent difficile à calculer. 3) Si $Q(X_i \to X_{i'}) = Q(X_{i'} \to X_i)$, cet algorithme est identique à l'algorithme orginal de Metropolis.

3.3.3 Recuit simulé

Le recuit simulé (simulated annealing) est une expérience réalisée par Metropolis *et al.* dans les années 1950 pour simuler l'évolution du processus de recuit physique [44]. Ce processus est utilisé en métallurgie pour améliorer la qualité d'un métal. On essaie de minimiser la taille des cristaux par des réchauffements et des refroidissements itératifs. En partant d'une haute température à laquelle la matière est devenue liquide, la phase de refroidissement conduit la matière à retrouver sa forme solide par une diminution progressive de la température.

L'idée est d'effectuer un mouvement selon une distribution de probabilité qui dépend de la qualité des différents voisins et d'un paramètre, appelé la température (note T) :

- T élevée : tous les voisins ont à peu près la même probabilité d'être acceptés.
- T faible : un mouvement qui dégrade la fonction de coût a une faible probabilité d'être choisi.
- $T = 0$: aucune dégradation de la fonction de coût n'est acceptée.

Algorithme 1 : l'algorithme de recuit simulé

```
1:  Procédure RECUIT_SIMULÉ
2:      n = 1 ; T = c ; X_i = initial
3:      tant que T ≥ ϵ faire
4:          choisir aléatoirement X_j ∈ Voisin_{X_i}
5:          si E(X_i) < E(X_j) alors
6:              X_i = X_j
7:          sinon
8:              u = Uniforme[0, 1]
9:              si u < e^{-(E(X_j)-E(X_i))/T} alors
10:                 X_i = X_j
11:             sinon
12:                 X_i reste inchangé
13:             fin si
14:         fin si
15:     fin tant que
16:     n = n + 1 ; T = c/ ln(n)
17: fin Procédure
```

Dans cet algorithme, ϵ est un réel proche de 0, $Voisin$ est une fonction qui retourne l'ensemble d'états voisins de l'état X_i.

La température varie au cours de la marche aléatoire : T est élevée au début, puis diminue

et finit par tendre vers 0. C'est une méthode importante historiquement et facile à implémenter, elle possède des propriétés de convergence intéressantes.

3.3.4 L'échantillonnage de Wang-Landau

La plupart des méthodes de Monte-Carlo, par exemple, l'échantillonnage suivant l'importance [44], l'algorithme de Swendsen-Wang [50], *etc.*, génèrent une distribution canonique de la température donnée $g(E)e^{-E/RT}$. Ces algorithmes sont limités au sens que nous devons lancer plusieurs fois l'algorithme si nous voulons connaître des quantités thermodynamiques sur un ensemble de température.

L'algorithme de Wang-Landau proposé par Fugao Wang et David P. Landau [51, 52] est une méthode de Markov Chaîne Monte Carlo (MCMC) avec histogramme . Il est conçu spécialement pour calculer une des quantités les plus importantes dans la physique statistique : la densité d'états d'énergie $g(E_i)$, qui est définie comme suit :

Définition 3.9 ***la densité d'états d'énergie,*** *$g(E_i)$ est le nombre de toutes les configurations possibles dans un intervalle d'énergie E_i d'un système.*

Définition 3.10 ***la densité relative d'états d'énergie,*** *$g(E_i)$ est le nombre de toutes les configurations possibles dans un intervalle d'énergie E_i divisé par le nombre de toutes les configurations d'un système.*

Si la densité d'états d'énergie ne dépend pas de la température, alors une fois que nous avons estimé la densité d'états pour toutes les énergies possibles, nous pouvons par exemple calculer la fonction de partition $Z = \sum_E g(E)e^{-E/RT}$. La plupart des quantités thermodynamiques, comme l'entropie, l'enthalpie, l'énergie libre, peuvent être obtenues à partir de la valeur de Z.

Cet algorithme a été initialement conçu pour les systèmes physiques. Il est basé sur une connaissance de la distribution de Boltzmann, c'est-à-dire qu'à une température donnée, les molécules sont réparties entre les hautes énergies, ou des états défavorables, et les faible énergies, ou des états favorables, avec une probabilité donnée par la différence d'énergie et les densités d'états (voir la section 2.2).

En principe, l'algorithme de Wang-Landau peut être appliqué à tout système qui se caractérise par une fonction de coût (ou d'énergie) . Par exemple, il a été appliqué à la solution des intégrales numériques [53] et au repliement des protéines [54].

Pour déterminer la densité d'états d'énergie en utilisant l'algorithme de Wang-Landau, nous définissons quelques notions (dont certaines sont déjà définies en section 3.2) :

- Soit $\Omega = E_1, ..., E_i, ...$ l'espace d'états d'énergie.
- Soit E_i un état d'énergie, il correspond à un ensemble de conformations ayant l'énergie dans l'intervalle de E_i.
- Soit G la densité d'états d'énergie. Soit $g(E_i)$ le nombre de conformations ayant l'énergie dans l'intervalle d'énergie de $[E_i, E_i + \Delta E]$.
- Soit H l'histogramme d'états d'énergie. Soit $h(E_i)$ le nombre de conformations échantillonnées en l'état E_i.
- Soit $F = f_0, f_1, ..., f_{final}$ les facteurs de modification utilisés pour mettre à jour les valeurs de $g(E_i)$.

Nous itérons les étapes suivantes :

1. **Initialiser les paramètres** :

   - Définir un espace d'états de l'énergie Ω :

     Avec l'algorithme de Wang Landau, nous voulons prédire la densité d'états d'énergie G d'un système en effectuant une marche aléatoire de façon uniforme sur l'ensemble d'états d'énergie. Soit R l'ensemble de conformations possibles du système. Soit E_R l'énergie de toutes ces conformations. Soit e_{min} et e_{max} l'énergie minimale et l'énergie maximale de E_R. L'espace d'états de l'énergie Ω peut être obtenu en découpant l'intervalle $[e_{min} : e_{max}]$ en un ensemble de sous-intervalles de la même taille δE. Chaque sous-intervalle de l'énergie $[E_i, E_i + \Delta E]$ correspond à un état.
     La figure 3.2 illustre un exemple où les valeurs de e_{min} et e_{max} sont -1 kcal/mol et 2 kcal/mol. Si nous décidons que la taille d'un sous-intervalle est 0.1 kcal/mol, alors nous aurons au total 30 états $(E_1, ..., E_{30})$ dans l'espace d'états de l'énergie Ω.

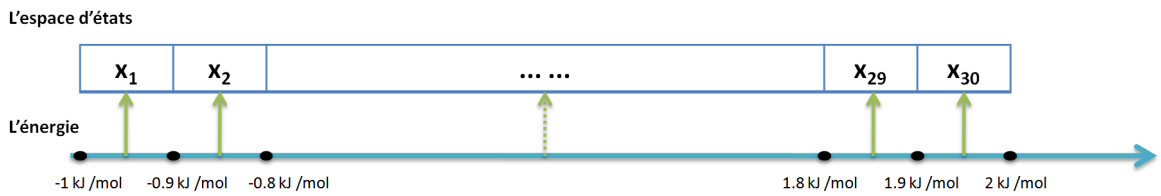


FIGURE 3.2 – Un exemple de l'espace de 30 états de l'énergie : $X_1, ..., X_{30}$.

   - choisir un état initial $E_0 \in \Omega$.

– définir la valeur du facteur initial de modification f_0 (par défaut, $f_0 = e$).

– initialiser toutes les valeurs de la densité d'états d'énergie G : $\forall E_i \in \Omega$, $g(E_i) = 1$.

– initialiser toutes les valeurs de l'histogramme de l'énergie H : $\forall E_i \in E$, $h(E_i) = 0$.

2. **Lancer l'algorithme de Wang Landau avec l'état initial E_0.**

Algorithme 2 : l'algorithme de Wang Landau

```
1:  Procédure WANG_LANDAU(E_0)
2:      initialiser G
3:      tant que f > 1 + ϵ faire
4:          initialiser H = 0
5:          tant que H n'est pas plat faire
6:              pour i = 0 jusqu'à NumSteps faire
7:                  proposer aléatoirement un nouvel état E_i' à partir de E_i
8:                  z ← random_double(0, 1) ;
9:                  si g(E_i') == 0 ou z < g(E_i)/g(E_i') alors
10:                     E_{i+1} ← E_i' ;
11:                 sinon
12:                     E_{i+1} ← E_i ;
13:                 fin si
14:                 mettre à jour H et G ;
15:             fin pour
16:         fin tant que
17:         f ← sqrt(f);
18:     fin tant que
19:     retourner G ;
20: fin Procédure
```

Dans l'algorithme 2,

– en ligne 3, ϵ est une constante prédéfinie, où $1 \gg \epsilon > 0$.

– en ligne 5, selon l'article [51], les auteurs définissent qu'un histogramme H est "plat" si $\forall E_i \in \Omega, h(E_i) \geq \eta \cdot moyenne(H), 1 \geq \eta > 0$.

– en ligne 6, $NumSteps$ est une constante prédifinie, qui permet de vérifier si l'histogramme H est plat après $NumSteps$ étapes de simulation de Wang Landau.

– en ligne 14, nous faisons la mise à jour par : $h(E_{i+1}) = h(E_{i+1})+1, g(E_{i+1}) = g(E_{i+1}) \cdot f$.

– en ligne 17, $sqrt$ est la fonction racine carrée.

3. **Récupérer la densité relative d'états d'énergie : G**

3.4 Méthode de RNA-WL

J'ai développé l'algorithme RNA-WL qui prend une séquence ou une hybridation de deux séquences d'ARN en entrée. Il va ensuite procéder à une marche aléatoire (ou une simulation de la chaîne de Markov) sur l'ensemble des structures secondaires de cette séquence ou hybridation d'ARN pour enfin produire une approximation de la densité relative d'états d'énergie de toutes ses structures secondaires.

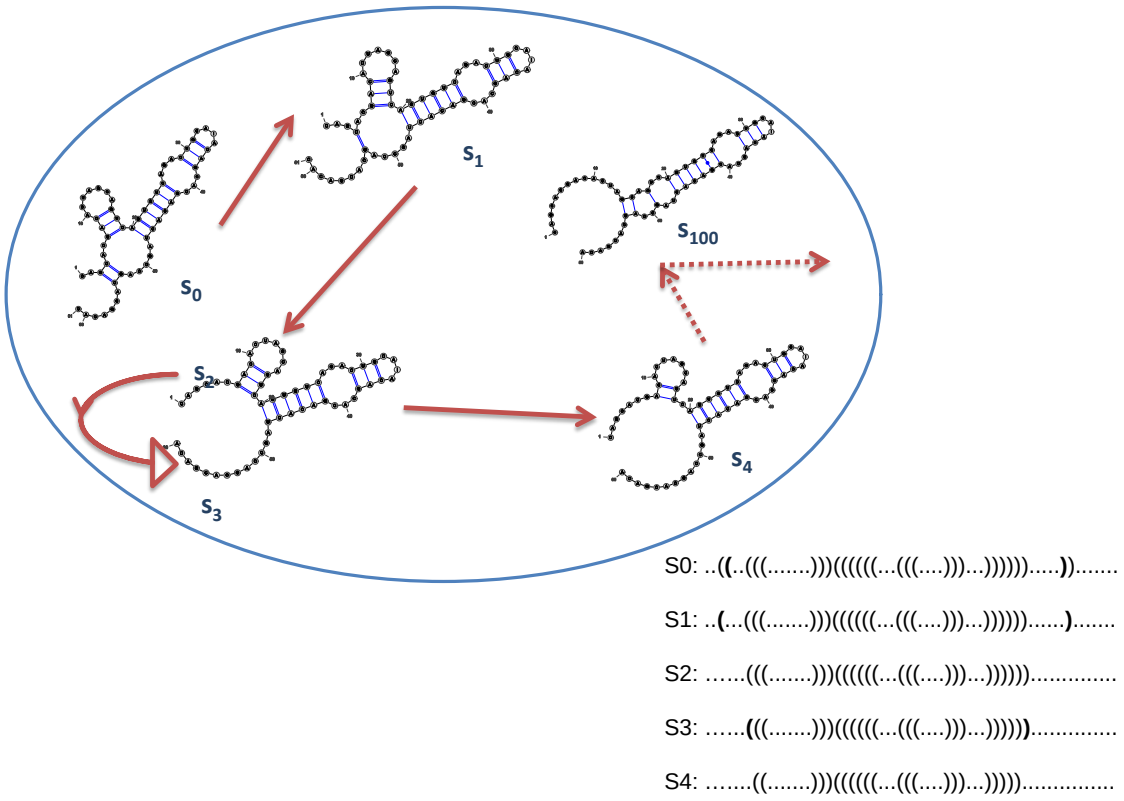


FIGURE 3.3 – L'illustration d'une simulation de RNA-WL, où s_0, s_1, ..., sont les structures secondaires à chaque étape de la simulation.

La figure 3.3 illustre une marche aléatoire possible sur la séquence "CAGCACGACACUAGCAGUCAGUGUCAGACUGCAAACAGCACGACACUAGCCAGCACGACAC". Dans cet exemple, la struture MFE[3] "..((..(((.....)))((((((...(((....)))...)))))).....)).....)....." est choisie comme la strucutre secondaire initiale s_0, la marche s'effectue dans l'ordre : s_0, s_1, s_2, s_2, s_3, ..., s_{100}, ...s_{fin}. Les transitions peuvent se produire entre deux structures secondaires différentes qui ne diffèrent que d'une paire de bases : les transitions $s_0 \rightarrow s_1, s_1 \rightarrow s_2, ...etc.$, ou sur la

3. La structure ayant l'énergie libre minimume.

même structure : la transition $s_2 \rightarrow s_3, ...etc.$.

3.4.1 Algorithme

- Soit w une séquence d'une molécule ou d'une hybridation de deux molécules d'ARN.
- Soit s une structure secondaire de w.
- Soit e_{min} le minimum d'énergie libre de w.
- Soit $f, \epsilon, NumSteps$ les paramètres définis par l'utilisateur.

Algorithme 3 : l'algorithme de `RNA-WL`

```
Procédure RNA-WL (w)
    initialiser G
    tant que f > 1 + ϵ faire
        intialiser H = 0
        tant que H n'est pas plat faire
            pour i = 0 jusqu'à NumSteps faire
                PROCHAINE_GÉNÉRATION(w, s, G, H)
            fin pour
        fin tant que
        f ← sqrt(f)
    fin tant que
fin Procédure
```

Algorithme 4 : l'algorithme de PROCHAINE_GÉNÉRATION

1: **Procédure** PROCHAINE_GÉNÉRATION(w, s, G, H)
2: proposer une structure secondaire $s_{next} \in V(s)$
3: $E_i \leftarrow bin(E[s])$
4: $E_{i'} \leftarrow bin(E[s_{next}])$
5: $z \leftarrow random_double(0, 1)$
6: **si** $g(E_{i'}) == 0$ ou $z < g(E_i)/g(E_{i'})$ **alors**
7: $s = s_{next}$
8: $E_i = E_{i'}$
9: **sinon**
10: s et E_i inchangées
11: **fin si**
12: $g(E_i) = g(E_i) \cdot f$
13: $h(E_i) = h(E_i) + 1$
14: **retourner** s, G, H
15: **fin Procédure**

Dans l'algorithme 4,

- en ligne 2, $V(s)$ est la fonction qui retourne l'ensemble des structures secondaires : S où $\forall s_i \in S, D_{bp}(s, s_1) = 1$ (voir la distance de paires de bases D_{bp} est définie dans la section 4.3).
- en ligne 4, $E(s)$ est l'énergie libre de s, $bin(E[s])$ indique l'état d'énergie de la structure s. Voici un exemple, si $E(s) = 1.23$ kcal/mol, $e_{min} = -10$ kcal/mol, l'énergie libre de s est dans l'intervalle $[-10 + (113 - 1) \cdot 0.1 : -10 + 113 \cdot 0.1]$ kcal/mol, alors $bin(E[s])$ indique que la structure s est dans l'état de l'énergie : E_{113}.

3.4.2 Simulation de la chaîne de Markov

Notre algorithme `RNA-WL` est une méthode de Monte Carlo par Chaînes de Markov (MCMC) sur l'ensemble des structures secondaires de la séquence d'ARN donnée.

À chaque étape de Monte Carlo, nous avons une structure secondaire actuelle s, et un ensemble de structures secondaires V parmi lesquelles nous allons ensuite proposer une nouvelle structure secondaire s_{next} avec une probabilité uniforme. Selon la règle de rejet (voir la ligne 5 de l'algorithme 3), soit nous acceptons de prendre la nouvelle structure secondaire $s \leftarrow s_{next}$, soit nous gardons la même structure secondaire $s \leftarrow s$. Après avoir mis à jour les valeurs de la

densité d'états d'énergie $g(E_i)$ et l'histogramme $g(E_i)$ où $e_{min}+(i-1)\cdot 0.1 \leq \mathrm{E}(s) < e_{min}+i\cdot 0.1$, nous passons à l'étape suivante. (voir la figure 3.4)

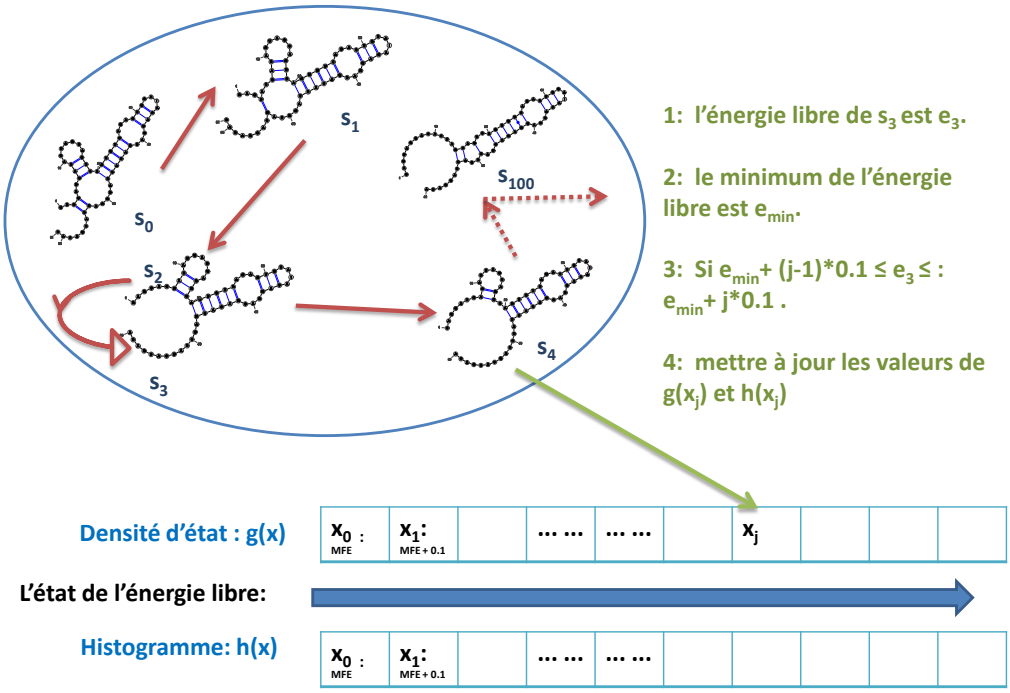


FIGURE 3.4 – La densité d'états d'énergie G et l'histogramme H utilisés dans la simulation de `RNA-WL`.

3.4.2.1 Densité d'états de l'énergie libre : $g(E_i)$

Au début de notre simulation, la densité d'états d'énergie est a priori inconnue, alors nous avons simplement mis les valeurs $g(E_i) = 1$ pour tous les états d'énergie possible E_i. Ensuite, nous commençons la marche aléatoire à partir de la structure secondaire s_0, et la probabilité de visiter les structures en l'état E_i est proportionnelle à l'inverse de la densité d'états : $1/g(E_i)$. Si E_i et E_j sont les deux états avant et après la transition, alors la probabilité de passer de l'état E_i à l'état E_j est :

$$p(i,j) = \min\{\frac{g(E_i)}{g(E_j)}, 1\} \tag{3.20}$$

À chaque fois que l'état de l'énergie X_j est visité, nous modifions systématiquement sa

densité d'états d'énergie par un facteur de modification $f > 1$, i.e. $g(X_j) \to g(X_j) \cdot f$. (En pratique, nous utilisons la formule $\ln[g(X_j)] \to \ln[g(X_j)] + \ln(f)$ pour éviter que la valeur de $g(X_j)$ soit trop grande.) Si la marche aléatoire rejette un mouvement possible à l'état X_j et reste à l'ancien état X_i, nous modifions également la densité d'états X_i avec le facteur de modification f. Dans [51], les auteurs utilisent le facteur de modification $f = e^1 \simeq 2,718$, qui nous permet d'atteindre tous les états de l'énergie possible très rapidement. Si f est trop petit, la marche aléatoire va prendre un temps extrêmement long pour atteindre tous les états de l'énergies.

3.4.2.2 Histogramme de l'énergie libre : $h(X_i)$ et Facteur de modification : f

L'histogramme H permet de compter le nombre de structures secondaires échantillonnées pour chaque état d'intervalle d'énergie X_i. Au début, les valeurs de l'histogramme H sont toutes initialisées à 0. À chaque fois qu'une structure secondaire de l'état de l'énergie X_i est échantillonnée, nous incrémentons la valeur de $h(X_i)$, i.e. $h(X_i) = h(X_i) + 1$.

Après avoir fini les $NumSteps$ étapes de la simulation de Wang Landau, on vérifie systématiquement que l'histogramme est "plat". Lorsqu'il est "plat", nous savons qu'à ce moment là, la densité d'états d'énergie est proche de la vraie valeur avec une précision proportionnelle à la valeur de $\ln(f)$ [51, 52]. Puis, nous réduisons le facteur de modification à une autre valeur plus fine en utilisant une fonction comme $f_{i+1} = \sqrt{f_i}$, réinitialisons l'histogramme, et recommençons la nouvelle simulation de Wang Landau avec le nouveau facteur de modification f_{i+1}. Nous continuons jusqu'à ce que l'histogramme soit "plat" à nouveau, puis nous réduisons le facteur de modification $f_{i+2} = \sqrt{f_{i+1}}$ et redémarrons. Nous arrêtons la simulation quand le facteur de modification est inférieure à une valeur prédéfinie. ($f \leq 1 + \delta, 0 \ll \delta \ll 1$).

Il est très clair que la suite des facteurs de modification $f_1, f_2, ..., f_i, f_{i+1}, ...$ agit comme un paramètre de contrôle important pour la précision de la densité d'états. Il détermine également le nombre d'étapes de Wang Landau nécessaires à la simulation. La précision de la densité d'états dépend non seulement des valeurs de f, mais aussi de nombreux autres facteurs, comme la complexité et la taille du système, le critère de l'histogramme "plat", et d'autres détails de l'implémentation de l'algorithme.

Il est en général impossible d'obtenir un histogramme parfaitement "plat". La valeur de m est choisie en fonction de la taille, de la complexité du système et de la précision souhaitée de la densité d'états d'énergie. La valeur de m peut être petite lorsque la taille du système est petite, et lorsque la précision souhaitée de la densité d'états d'énergie n'est pas grande. La valeur de m peut être grande (par exemple : $m = 95$) lorsque la taille du système est grande, et lorsque

la précision souhaitée de la densité d'états d'énergie est grande, le temps nécessaire pour finir la simulation est long. Donc, il faut trouver un compromis entre la précision souhaitée de la densité et le temps d'exécution de la simulation.

3.4.3 Condition d'équilibre

Les articles [55, 56, 57] utilisent des algorithmes de Metropolis-Hastings pour déterminer la structure ayant le minimum d'énergie d'un biopolymère (ADN, ARN, protéine), dont la probabilité de se déplacer de l'état x_i à l'état x_j est donnée par :

$$\begin{aligned} p_{i,j} = P(X_i \to X_j) &= min(1, \frac{e^{-E(s_j)/RT}}{Z} / \frac{e^{-E(s_i)/RT}}{Z}) \cdot \frac{1}{N(s_i)} \\ &= min(1, e^{(E(s_i)-E(s_j))/RT}) \cdot \frac{1}{N(s_i)} \end{aligned}$$

où, s_i est la configuration du biopolymère à l'état X_i, $N(s_i)$ est le nombre de configurations qui peuvent être obtenues avec un mouvement à partir de s_i.

Dans le cas de la structure secondaire d'ARN, les états de la chaîne de Markov sont les intervalles de l'énergie libre des structures secondaires d'ARN. Les mouvements autorisés sont l'ajout ou la suppression d'une paire de bases [58].

Bien que la condition d'équilibre soit vérifiée pour l'algorithme de Metropolis-Hastings, ce n'est pas une condition nécessaire pour la convergence à la distribution stationnaire. Alors que dans notre cas de la structure secondaire d'ARN, la condition d'équilibre n'est pas toujours vérifiée.

Voici un exemple sur la séquence d'ARN : GGGGGCCCCC, nous sommes en l'état ayant la structure vide s_i = ... La structure s_i a 18 structures voisines, dont l'une est la structure s_j = (.....). La structure s_j a 11 structures voisines, dont l'une est la structure vide s_i.

Les énergies libres de ces deux structures sont : $E(s_i) = 0\ kcal/mol$, $E(s_j) = 2,70\ kcal/mol$.

La fonction de partition dépend de la température T. Ici, $T = 310°$C, ce qui donne $Z = 621,5$. La constante universelle des gaz parfaits : R = 0.001987 $kcal \cdot mol^{-1} \cdot K^{-1}$.

Les probabilités stationnaires de ces deux états sont : $p_i^* = \frac{1}{621,5} = 0,00161$, $p_j^* = \frac{0,012456}{621,5} = 0,00002$.

Les probabilités de transition entre ces deux états sont : $p_{i,j} = \frac{0,012456}{18} = 0,00692, p_{j,i} =$

$\frac{1}{11} = 0,09091$, alors, nous avons :

$$\begin{aligned} p_i^* \cdot p_{i,j} &= 0,00161 * 0,00692 = 692,01 \times 10^{-6} \\ p_j^* \cdot p_{j,i} &= 0,00002 * 0,09091 = 1,82 \times 10^{-6} \end{aligned}$$

Comme $p_i^* \cdot p_{i,j} \neq p_j^* \cdot p_{j,i}$, alors la condition d'équilibre n'est pas vérifiée.

Pour le même exemple, si nous considérons que chaque structure secondaire est la structure voisine de toutes les structures secondaires, c'est-à-dire que $\forall (s_i, s_j) \in S$ et $s_i \neq s_j$, $D(s_i, s_j) = 1$, où S est l'ensemble des structures secondaires de la séquence d'ARN donnée.

$$\begin{aligned} p_{i,j} = P(E_i \to E_j) &= min(1, \frac{e^{-E(s_j)/RT}}{Z} / \frac{e^{-E(s_i)/RT}}{Z}) \cdot \frac{1}{N} \\ &= min(1, e^{(E(s_i)-E(s_j))/RT}) \cdot \frac{1}{N} \end{aligned}$$

où, N est le nombre de structures secondaires dans S.

Dans ce cas là, les chaînes de Markov sont réversibles, la condition d'équilibre est vérifiée. Malgré le caractère non réversible des chaînes de Markov de ces algorithmes de Metropolis, dont les états sont les structures secondaires d'une séquence donnée d'ARN, et dans lesquels on se déplace en ajoutant ou supprimant une seule paire de bases, ce type d'algorithme sans la condition d'équilibre est appliqué couramment dans les articles [58, 59, 60, 61]. Pour cette raison, nous n'hésitons pas à appliquer l'algorithme de Wang-Landau pour l'étude des structures secondaires d'ARN.

3.4.4 Densité d'états de l'énergie

Notre algorithme `RNA-WL` (voir les algorithmes 3 et 4) calcule en fait la densité relative d'états d'énergie. Pour obtenir la densité d'états d'énergie, nous aurons besoin du nombre total des structures d'un ARN N.

Nous pouvons calculer la valeur de N par un programme dynamique, décrit comme suit. Etant donnée une séquence w d'ARN de longueur n, soit θ le nombre minimum de nucléotides non appariés dans une boucle terminale. Rappelons que $BP_{i,j} = 1$ si les nucléotides à la position i et j peuvent former une paire de base Watson-Crick ou Wobble, sinon, $BP_{i,j} = 0$, $N_{i,j}$ le

nombre de structures secondaires de la sous séquence $w[i,j]$. Nous avons donc :

$$\begin{aligned} N_{i,j} &= 0, \ si \ j < i+3 \\ N_{i,j} &= N_{i,j-1} + \sum_{k=i}^{j-\theta-1} BP_{k,j} \cdot N_{i,k-1} \cdot N_{k+1,j-1}, \ sinon \end{aligned}$$

Algorithme 5 : l'algorithme pour compter le nombre de structures secondaires sans pseudo-noeuds d'une séquence d'ARN

1: **Procédure** N1(rna)
2: $m = \text{len}(rna)$
3: **si** $m \leq 0$ **alors**
4: **retourner** 0
5: **sinon si** $0 < m \leq \Theta + 1$ **alors**
6: **retourner** 1
7: **sinon**
8: r = N1(rna[1 :m-1])
9: r += $\sum_{k=1}^{m-\Theta-1} BP(rna[k], rna[m]) \cdot N1(\text{rna}[1:k]) \cdot N1(\text{rna}[k+1:m-1]))$
10: **retourner** r
11: **fin si**
12: **fin Procédure**

Le nombre total de structures est $N_{1,n}$. À partir de la densité relative d'états d'énergie obtenue par notre programme `RNA-WL`, nous pouvons calculer la densité d'états d'énergie par :

$$g(E_i) = g_{relative}(E_i) \cdot N \tag{3.21}$$

Pour une température donnée T, nous rappelons que la fonction de partition pour les énergies de toutes les structures secondaires est définie par :

$$Z(T) = \sum_{s \in S} e^{\frac{-E(s)}{RT}} \tag{3.22}$$

Dans le cas des structures secondaires d'ARN, la valeur de $Z(T)$ peut être calculée par la formule suivante :

$$Z(T) = \sum_{E_i \in \omega} g(E_i) \cdot e^{\frac{-E_i}{RT}} \tag{3.23}$$

Dans les articles [51, 52], les auteurs ont appliqué l'algorithme de Wang Landau dans le cas du modèle d'Ising. Le point fort de leur formule (la formule 2.9) est qu'elle permet de calculer pour une même densité d'états d'énergie la fonction de partition à n'importe quelle température désirée, si la densité d'états ne dépend pas de la température. Malheureusement, ce n'est plus le cas pour prédire l'énergie de structure secondaire d'ARN avec les paramètres du modèle de Turner [35], puisque les paramètres d'énergie libre pour des paires de bases empilées, des boucles terminales, des renflements, des boucles internes, *etc.*, dépendent tous de la température. Du coup, chaque exécution de notre programme `RNA-WL` peut seulement prédire la fonction de partition pour une température donnée.

3.5 Résultats

3.5.1 Description du programme `RNA-WL`

J'ai implémenté le programme `RNA-WL` en C. Ce programme est disponilbe sur le site d'internet : *http ://sourceforge.net/projects/rna-wl/*.

La figure 3.5 illustre une partie des résultats obtenu par notre programme `RNA-WL`, où la séquence d'ARN w = CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUU, la structure secondaire à l'état initial s_0 = la structure MFE de w, f = le nombre exponentiel : e, $NumSteps = 1000$, $m = 90$. Après avoir échantillonné 17402000 structures secondaires de w, une approximation de la densité d'états d'énergie est obtenue comme suit.

Turnenr energy	Relative Frequency	Secondary Structure
-3.3	5.451712319e-24	((.(((((....))))))).....
-3.0	5.9785285e-21	((.(((((....)))))))......(((.....)))
-2.8	1.462897691e-18	...(((((....))))).....
-2.5	1.095005691e-22	...(((((....)))))........(((.....)))
-2.2	8.631228828e-47	...(((((....))))).(((.....)))....
-2.1	1.170175048e-09	.....((((..(((.....)))..))))
-1.9	5.9785285e-21	((.(((((....)))))))).....((.....)).
-1.7	9.856927105e-21	((.(((((....)))))))).....(....).....
-1.6	4.24204945e-43	..((..(((((.....))))).)).
-1.5	4.907475037e-22	..((..(((((.(.....).))))).)).
-1.4	6.556252591e-18	...(((((....)))))).....((.....)).
-1.2	3.26416596e-19	((.((((.....))))))).....(((.....)))
-1.1	4.13379381e-33	...((((.....)))).....
-1.0	1.095005691e-22	...((((.....)))).....
-0.9	5.244360724e-09	.....((((.....))))....

.....

FIGURE 3.5 – L'illustration d'une partie de la sortie du programme `RNA-WL`, où la séquence d'ARN w = CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUU

Dans cette figure,

- la première colonne contient les énergies libres du modèle de Turner, chaque valeur d'énergie e_i correspond à un état x_i.
- la deuxième colonne contient la densité relative d'états d'énergie g_i, soit S_i l'ensemble des structures secondaires dont leur énergies sont dans un intervalle autour de l'énergie e_i.
- la troisième colonne contient les structures ayant le minimum d'énergie libre parmi l'ensemble de structures S_i.

Notre programme `RNA-WL` permet à l'utilisateur de modifier la taille de l'intervalle d'énergie, la taille par défaut est de 0,1 kcal/mol qui est aussi la précision maximum du modèle de Turner ; Les intervalles vides, où aucune structure n'a pas été échantillonnée, ne sont pas affichés. Dans cette exemple, la structure MFE échantillonnée par notre programme est ((.(((((....)))))))..... avec l'énergie libre à -3.3 kcal/mol, qui est identique à la structure MFE obtenue par l'autre programme RNAfold[38]. Seulement une partie des résultats pour l'intervalle d'énergie de -3.3 kcal/mol à -0.2 kcal/mol est affichée.

3.5.2 Prédiction de la densité d'états d'énergie pour une molécule d'ARN

3.5.2.1 Validation

Le programme `RNAsubopt` [62, 63] lit des séquences d'ARN en entrée, et ensuite calcule toutes les structures secondaires sous-optimales au sein d'une gamme d'énergie au-dessus de la MFE définiees par l'utilisateur. Lorsqu'il est utilisé avec l'option -p, `RNAsubopt` produit des échantillonnages de structures secondaires pondérées avec la probabilité de Boltzmann. Avec l'option -D[62], `RNAsubopt` calcule aussi la densité absolute d'états d'énergie de toutes les strucutres secondaire de la séquence d'ARN donnée, puis calculer la fonction de partition exacte d'énergie.

Nous avons pris une séquence d'ARN : ACCUGGCUGGGGGUAUCUCGUGAUCAUGAAGACGGGAUCCCCAUGGUGA pour tester si la densité d'états d'énergie obtenu par la simulation de `RNA-WL` est identique à la densité exacte d'états d'énergie calculée par le programme `RNAsubopt`. Dans la figure 3.6, on observe que la densité estimée par la simulation de `RNA-WL` est proche de la densité exacte obtenue par `RNAsubopt`.

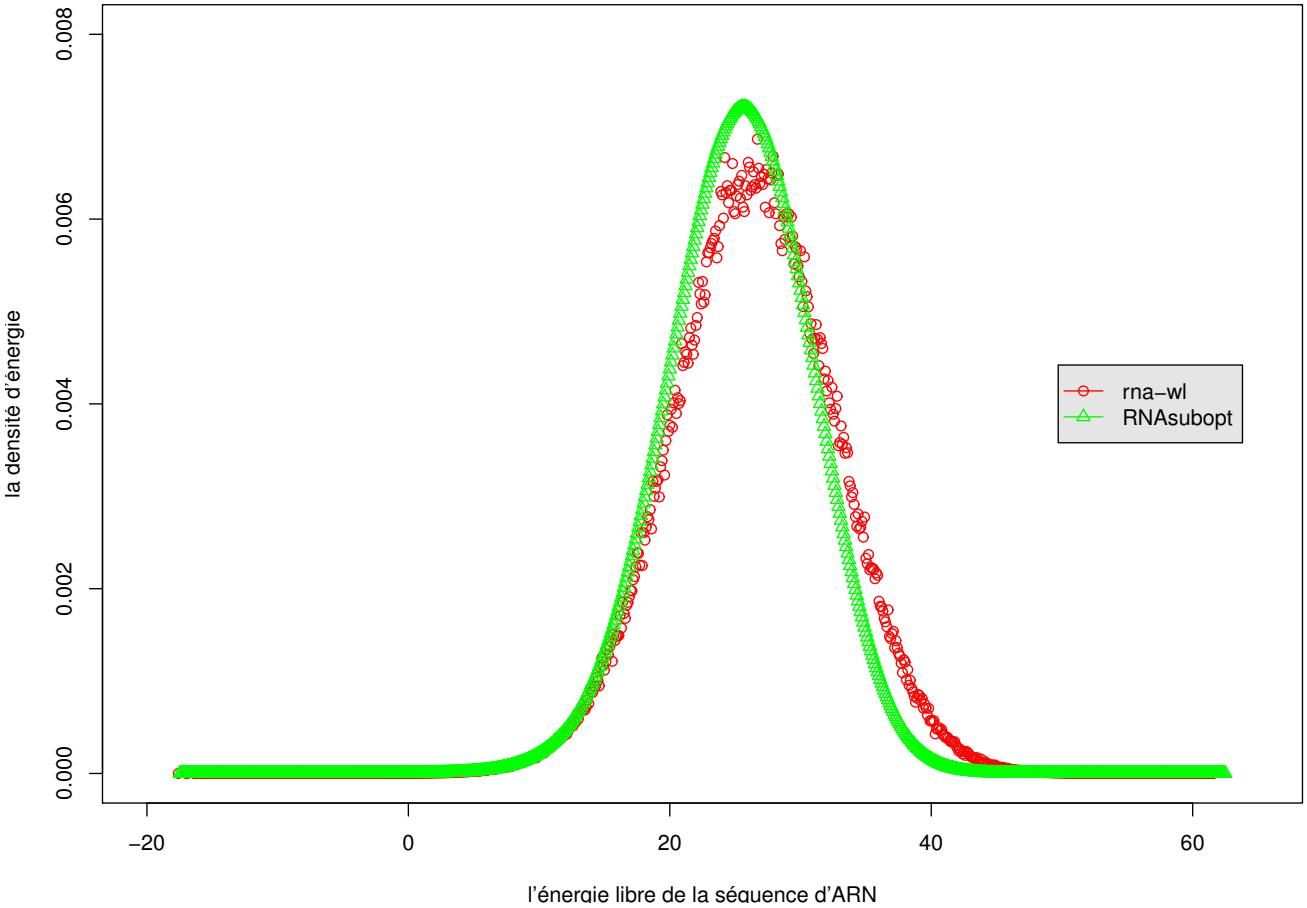


FIGURE 3.6 – La validation de la densité d'états d'énergie estimée par notre programme `RNA-WL` (courbe rouge) avec celle du programme `RNAsubopt` (courbe vert).

Pour savoir plus précisément la qualité de la densité estimée par notre programme `RNA-WL`, nous choisissons donc 24 séquences réelles et artificielles d'ARN de tailles 36 nucléotides jusqu'à 136 nucléotides, nous effectuons ensuite les tests du χ^2 pour comparer les deux densités d'états d'énergie obtenues par `RNA-WL` et `RNAsubopt`. Voici 8 des 36 séquences, les distances de test de χ^2, leur valeurs critiques et les superpositions des distributions :

1 : X00063.1/1061-1096 : CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUU

2 : L00073.1/390-426 : CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUUC

3 : CCGCGGAGGUCCGGCUAUGGGAAAUACCAAAAAAGC

4 : AAGACAUAGGAAAUACGUAGGGACAAGUGACUAACA

5 : AB010982.1/1-45 : AUGAACAACCAACGAAAAAGGACGGGAAAACCGUCUAUCAAUAUG

6 : AY152108.1/1-42 : AUGAACCAACGAAAAAAGGUGGUUAGACCACCUUUCAAUAUG

7 : AAUAAGCGAAAAAAGAAACACCGCAAAAAAUCAUCAUAGCAAAAC

8 : AAGUCAGCGGGGAGGCAACACACUGGGAAACUCAUCAAUGGA

Indice	Réelle / Artificielle	Distance de χ^2	Valeur critique
1	Réelle	0.0252	36.42
2	Réelle	0.0280	40.11
3	Artificielle	0.3354	36.42
4	Artificielle	0.6940	30.14
5	Réelle	0.0743	36.42
6	Réelle	0.1403	38.89
7	Artificielle	0.1847	28.87
8	Artificielle	0.3151	41.34

TABLE 3.1 – Les informations des séquences qui sont utilisées pour les tests du χ^2.

Le test du χ^2 est utilisé ici comme un test d'adéquation, il s'agit alors de se demander si les deux listes de valeurs de même effectif sortant du programmme `RNA-WL` et du programmme `RNAsubopt` peuvent dériver de la même loi de probabilité. La table 3.1 montre une partie des résultats, dont la première colonne indique les indices des séquences, la deuxième colonne indique si les séquences sont réelles ou artificielles, la troisième colonne présente les distances de χ^2 et la dernière colonne indique les valeurs critiques quand la valeur p est égale 0.05. Si la distance de χ^2 est inférieure à sa valeur critique, alors, le fait que les deux distributions suivent la même loi de probabilité est statistiquement signficatif. Plus la distance de χ^2 est petite, plus les deux densités d'états d'énergie s'approchent. La figure 3.7 illustre les superpositions des densités obtenues du programme `RNA-WL` et du programme `RNAsubopt`.

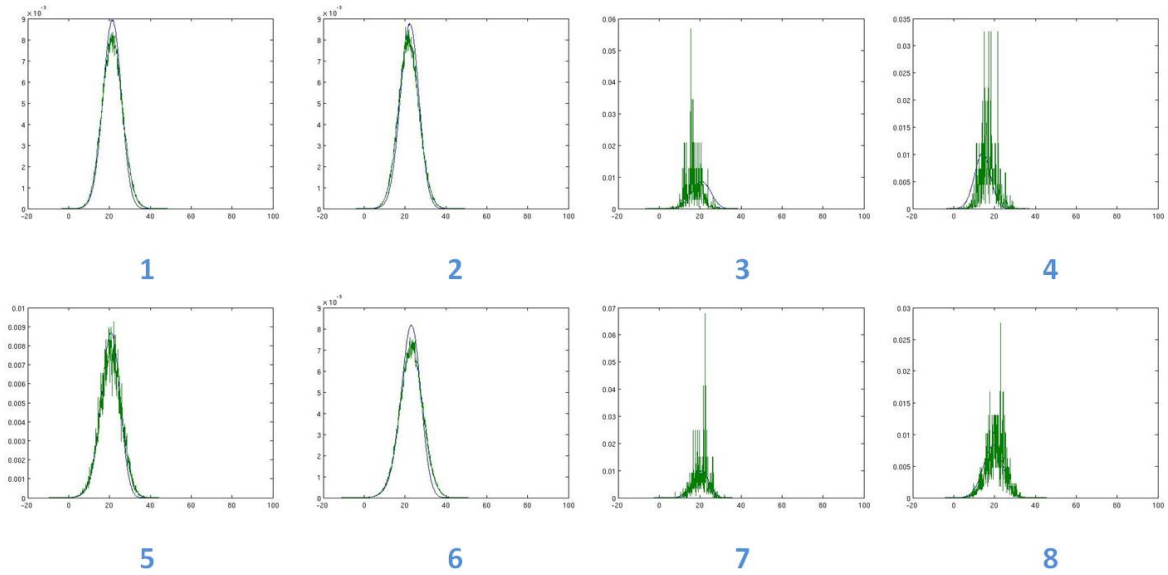


FIGURE 3.7 – Les superpositions des densités d'états d'énergie obtenues par notre programme `RNA-WL` (courbe vert) et par le programme `RNAsubopt` (courbe noir), sur les 8 séquences de longeur de 36 à 45 nucléotides.

À partir des résultats dessus, nous avons observé que 1 : les distances de χ^2 sont toutes petites et inférieures aux valeurs critiques correspondantes. 2 : les distances de χ^2 des séquences réelles sont toutes plus petites par rapport celles des séquences artificielles, la courbe de `RNA-WL` des séquences réelles s'approche plus de la courbe de `RNAsubopt` par rapport celle des séquences artificielles. Nous pouvons donc dire qu'il n'y a pas de différence significative entre la densité d'états d'énergie estimée par notre programme `RNA-WL` et la vraie densité obtenue par le programme `RNAsubopt`, l'estimation de la densité d'états d'énergie sur des séquences reélles d'ARN s'approche plus de la vrai densité par rapport aux séquences artificielles.

3.5.2.2 Diversité structurale

Nous allons maintenant étudier la diversité de toutes les structures échantillonnées par notre programme `RNA-WL`. Nous voulons savoir si notre échantillonnage couvre potentiellement toutes les structures secondaires d'ARN ou seulement une partie de structures secondaires. Comme le programme `RNAsubopt` permet aussi d'échantillonner des structures secondaires d'ARN par la probabilité de Boltzmann, nous comparons alors les diversités de ces deux ensembles de structures échantillonnées. Voici les 5 séquences que nous avons utilisé pour échantillonner des structures secondaires (Ces séquences sont données par Robert Giegerich) :

- **Les séquences utilisées :**

  >**fdhA -16 :** CGCCACCCUGCGAACCCAAUAAUAAAAUAUACAAGGGAGCAAGGUGGCG
  >**formyl-MFR +115 :** AUGUUGGAGGGGAACCCUGUAAGGGACCCUCCAACAU
  >**fruA +40 :** CCUCGAGGGGAACCCGAAAGGGACCCGAGAGG
  >**hdrA +105 :** GGCACCACUCGAACCGUAACGGAAAGUGGUGCU
  >**selD +67 :** UUACGAUGUGCCGAACCCUUUAAGGGAGGCACAUCGAAA

- **Ensemble Free Diversity** $= \sum_{i=0}^{N}\sum_{j=0}^{N} P_{(i,j)} \cdot (1 - P_{(i,j)})$.

  Cette diversité est proposée par le Vienna Package [64] , elle calcule à partir des probabilités que les nucléotides à la position i et j s'apparient $P_{(i,j)}$. La table 3.2 montre que les valeurs de "ensemble free diversity" du programme `RNA-WL` sont tous plus élevées par rapport à celles du programme `RNAsubopt` en température 37 °C et au point critique, ce qui signifie que notre programme `RNA-WL` produit un ensemble des structures secondaires

ID	RNA-WL 37 °C	RNA-WL Tc	RNAsubopt 37 °C	RNAsubopt Tc
fdhA -16	16.67	14.92	2.57	4.67
formyl-MFR +115	14.15	13.10	1.38	4.07
fruA +40	10.76	9.86	0.31	5.11
hdrA +105	11.42	10.93	1.38	4.90
selD +67	14.27	13.42	1.13	5.24

TABLE 3.2 – Les "ensemble free diversity" calculées à partir des structures secondaires générés par le programme RNA-WL et RNAsubopt en température par défaut (37 °C) et au point critique(Tm) qui est calculé par le programme RNAheat include dans le Vienna Package.

plus variées que le programme RNAsubopt.

– **Mean Boltzman Weighted Distance** $= N - \sum_{i=0}^{N} \sum_{j=0}^{N} P^2_{(i,j)}$.

Identifiant de la séquence	RNA-WL	RNASubopt	Taille de la séquence
fdhA -16	47.96	24.53	49
formyl-MFR +115	35.57	10.90	37
fruA +40	30.90	12.44	32
hdrA +105	31.78	12.07	33
selD +67	38.10	15.11	39

TABLE 3.3 – Les "mean weighted Boltzmann distance"[65] calculées à partir des structures secondaires générées par le programme RNA-WL (la deuxième colonne) et RNAsubopt (la troisième colonne) en température par défaut (37 °C), et aussi la taille de séquence(la quatrième colonne).

Les valeurs de "mean Boltzman weighted distance" sortant du programme RNA-WL sont toutes plus proches de la longueur des séquences par rapport à celles du programme RNAsubopt. Cela signifie à l'instar du test précédent que notre programme RNA-WL produire un ensemble des structures plus variées, plus représentatives de toute la population des structures secondaires de la séquence d'ARN donnée que RNAsubopt.

3.5.2.3 Loi Normale ou distribution des valeurs extrêmes ?

Après avoir observé les densités d'états d'énergie de certains séquences d'ARN, nous nous posons une autre question : est-ce que la distribution d'états d'énergie s'approche plus d'une loi normale ou à la distribution des valeurs extrêmes ? Pour répondre à cette question, nous avons choisi 8 séquences d'ARN comme jeu d'essai. Les codes d'accès au site EMBL de ces 8 séquences sont : AB010982, AE008853, AE013612, AJ532311, AJ532513, BC056833, L00073 et X00063.

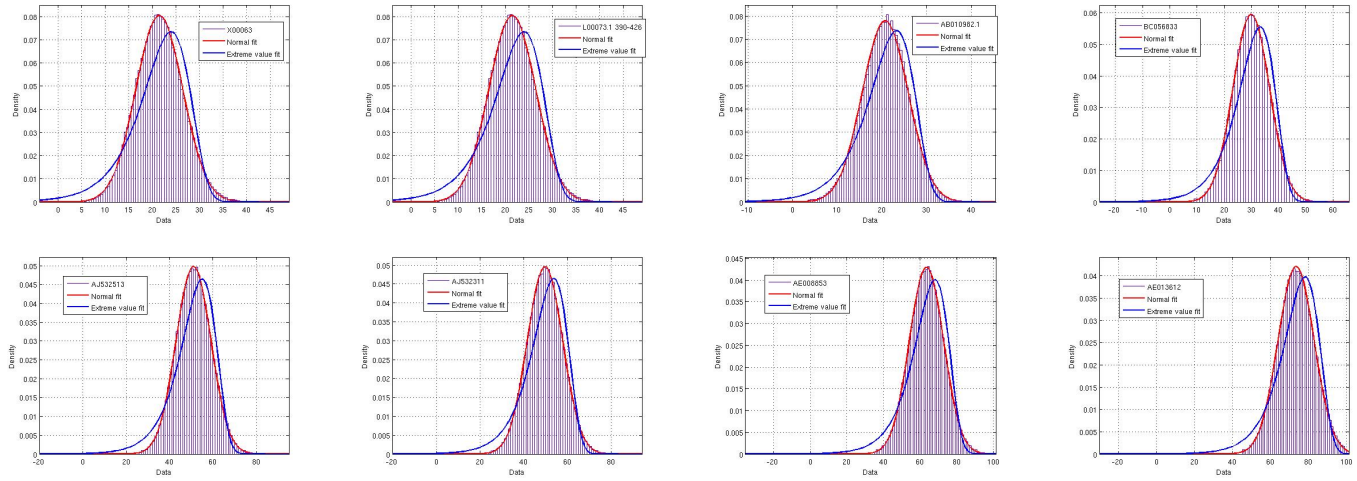


FIGURE 3.8 – L'ajustement de courbe de la distribution d'états d'énergie de `RNA-WL` (courbe violete) avec la loi normale (courbe rouge) et la distribution des valeurs extrêmes (courbe bleu).

La figure 3.8 présente la densité d'états d'énergie obtenue par `RNA-WL` (courbe violete) avec la loi normale (courbe rouge) et la distribution des valeurs extrêmes (courbe bleu) les mieux ajustées. Ces figures sont toutes générées par le programme Matlab.

La figure 3.9 montre la valeur efficace [4] de la densité de `RNA-WL` avec la loi normale la mieux ajustée (courbe bleu) et avec la distribution des valeurs extrêmes la mieux ajustée (courbe rouge).

Ces deux figures montrent que la densité relative d'états d'énergie des structures secondaires d'ARN s'approche plus d'une loi normale par rapport à la distribution des valeurs extrêmes. En plus, dans [66], il est rigoureusement démontré que la densité d'états d'énergie est asymptotiquement normale. Plus précisément, il est montré que si la longeur de séquence n tend vers l'infini, la densité relative d'états d'énergie d'une séquence d'ARN de longueur n est normale, où

4. "Root Mean Square" en anglais.

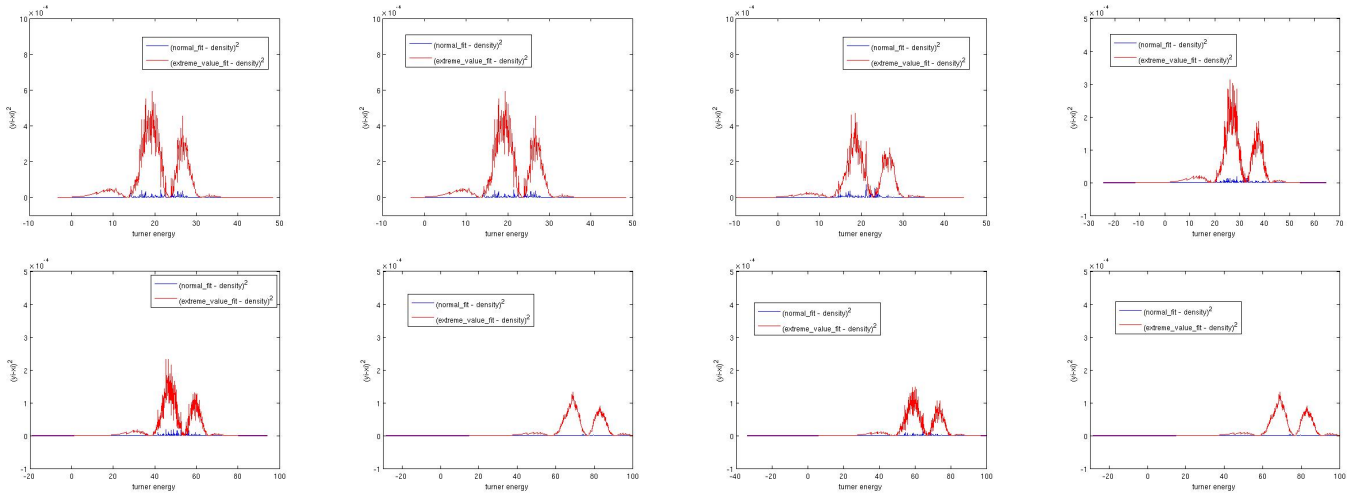


FIGURE 3.9 – (normal(X_i) - `RNA-WL`-densité(X_i))^2 : courbe bleu, (extremeValue(X_i) - `RNA-WL`-densité(X_i))^2 : couleur rouge

dans cet analyse mathématique, nous supposons que n'importe quelle base peut être appariée avec une autre base (le modèle homopolymère) et l'énergie d'une structure secondaire s est -1 fois le nombre de paires de base dans cette structure s (modèle d'énergie de Nussinov[34]).

3.5.2.4 Temps d'exécution.

Précédemment, nous avons montré que notre programme `RNA-WL` peut produire une bonne estimation de la densité d'états d'énergie. D'autre part, nous savons que le programme `RNAsubopt` peut énumérer toutes les structures secondaire seulement pour des séquences de petite taille. Nous nous intéressons donc à savoir la performance de notre programme `RNA-WL` pour des séquences de plus grande taille.

Nous avons choisi 40 séquences de tailles variés de 35 jusqu'à 136 nucléotides, nous exécutons les programmes `RNA-WL` et `RNAsubopt` sur toutes ces 40 séquences et enregistrons leur temps d'exécution.

La figure 3.10 montre les temps d'exécution de ces deux programmes. Chaque point correspond à la moyenne des temps d'exécution de 5 séquences d'un même intervalle de longueur. Nous avons bien observé que notre programme `RNA-WL` possède l'avantage de rapidité pour la prédiction de la densité d'états d'énergie des structures secondaires d'ARN par rapport l'autre programme `RNAsubopt`. Notre programme `RNA-WL` s'exécute plus vite que le programme `RNAsubopt` pour les séquences ayant plus de 45 nucléotides. Il peut traiter des séquences de longueur jusqu'à 136 nucléotides, contre 60 nucléotide pour le programme `RNAsubopt`.

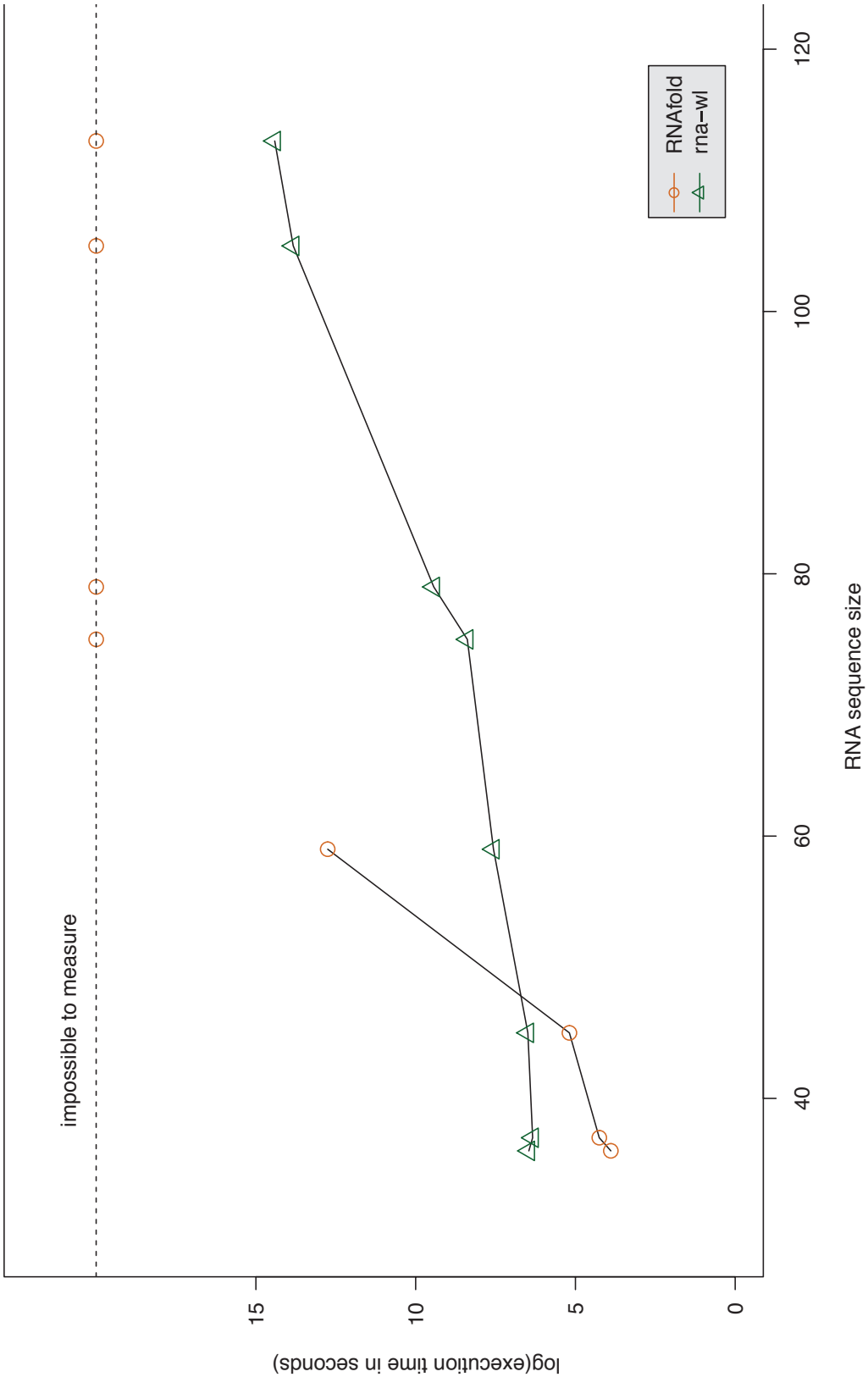


FIGURE 3.10 – log(temps d'exécution) en fonction de la taille de séquence du programme `RNA-WL` (point orange) et `RNAsubopt` (triange vert). L'ensemble de données utilisées pour tracer cette figure est présentées dans la section Annexe.

3.5.3 Prédiction de la température de dénaturation pour l'hybridation de deux molécules d'ARN

Dimitrov et Zuker [67] ont présenté une approche statistique pour décrire la procédure de l'auto-repliement avec hybridation de deux molécules d'ADN ou d'ARN A et B. Cette méthode prend en compte toutes les conformations possibles de l'espèce de simple et double brins en solution. En l'état d'équilibre, les cinq espèces vont être explorés : les monomères A et B, les homodimères 'AA' et 'BB' et l'hétérodimère 'AB'.

Cette méthode utilise les fonctions de partition de cinq espèces $Z_A, Z_B, Z_{AA}, Z_{BB}, Z_{AB}$ pour déterminer leur nombres de molécule $N_A, N_B, N_{AA}, N_{BB}, N_{AB}$ en l'état d'équilibre, et enfin obtenir ses quantités thermodynamiques, comme : l'énergie libre d'ensemble $\triangle G$, l'entropie $\triangle S$, l'enthalpie $\triangle H$, la capacité de chaleur C_p et la température de dénaturation T_m.

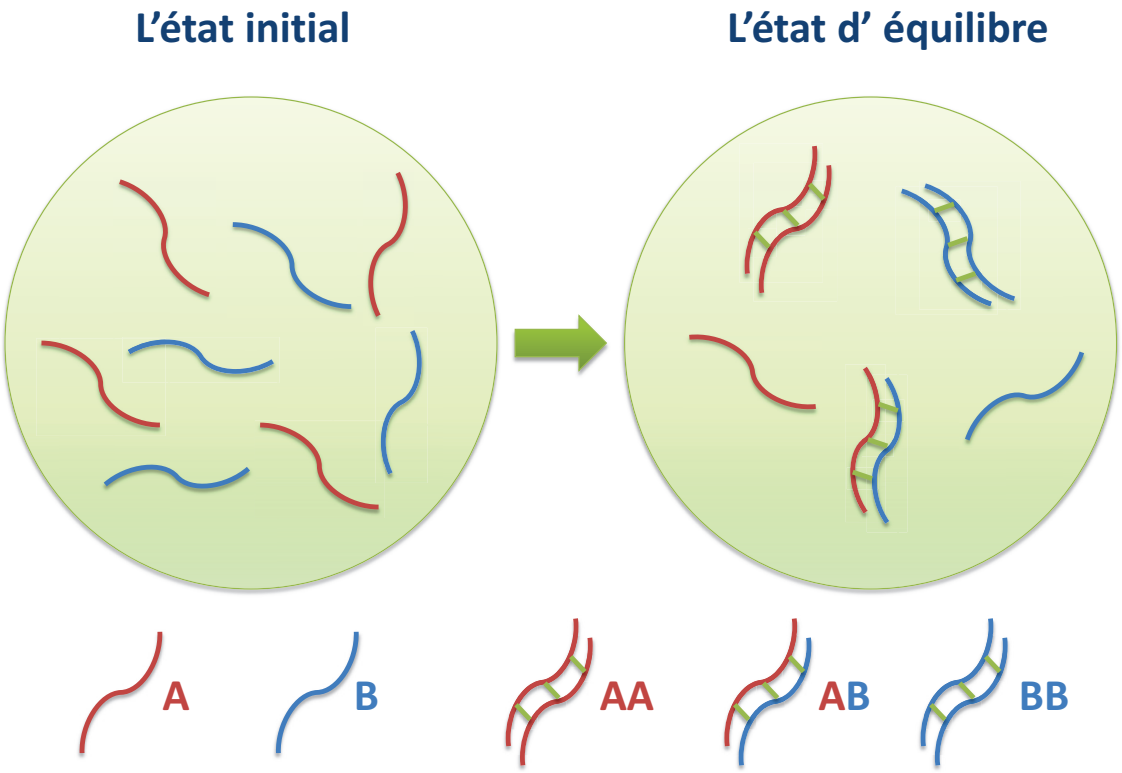


FIGURE 3.11 – L'ensemble des cinq espèces possibles : les molécules A et B, les hybridations de A-A, B-B, A-B.

Nous présentons ici un pipeline qui utilise les fonctions de partition calculées avec les densités d'états d'énergie de `RNA-WL`, et calcule la capacité de chaleur et la température de dénaturation de l'ensemble de cinq espèces à l'état équilibre.

3.5.3.1 Pipeline pour calculer la capacité de chaleur et la température de dénaturation

1. Compter le nombre de structures pour chacune des cinq espèces :
   - Soit $seqA$, $seqB$ les deux séquences d'ARN.
   - Soient i, j les positions de nucléotide des séquences $seqA$ et $seqB$.
   - l'algorithme 5 permet de compter le nombre de structures secondaires de $seqA$: $N(A)$ et de $seqB$: $N(B)$
   - l'algorithme 7 permet de compter le nombre d'hybridations entre les molécules A et A : $N(AA)$, B et B : $N(BB)$, A et B : $N(AB)$.

Algorithme 6 : l'algorithme pour compter le nombre d'hybridations uniquement avec des paires de bases de deux séquences d'ARN.

1: **Procédure** N1($seqA$,$seqB$, i, j)
2: **si** $i \leq 0$ or $j \leq 0$ **alors**
3: **retourner** 0
4: **sinon si** $i == 1$ **ou** $j == 1$ **alors**
5: **si** $i == 1$ **et** $j == 1$ **alors**
6: **retourner** BP(i, j) + 1
7: **sinon si** $i == 1$ **et** $j! = 1$ **alors**
8: **retourner** $1 + \sum_{k=1}^{j} BP(1, k)$
9: **sinon si** $i! = 1$ **et** $j == 1$ **alors**
10: **retourner** $1 + \sum_{k=1}^{i} BP(k, 1)$
11: **fin si**
12: **sinon**
13: **retourner** BP(1, j) + BP(1, i) + (BP(i, j) + 1) · N(seqA, seqB, i-1, j-1) + $\sum_{k=2}^{i-1} BP(k, j) \cdot N(seqA, seqB, k-1, j-1) + \sum_{k=2}^{j-1} BP(i, k) \cdot N(i-1, k-1)$
14: **fin si**
15: **fin Procédure**

Algorithme 7 : l'algorithme pour compter le nombre d'hybridations sans pseudo-noeuds de deux séquences d'ARN

1: **Procédure** NAB($seqA$,$seqB$, n, m)
2: **si** $n \leq \Theta + 1$ **et** $m \leq \Theta + 1$ **alors**
3: **si** $n == 0$ **ou** $m == 0$ **alors**
4: **retourner** 1
5: **sinon**
6: **retourner** N(seqA, seqB, n, m)
7: **fin si**
8: **sinon si** $n \leq \Theta + 1$ **et** $m > \Theta + 1$ **alors**
9: **retourner** NAB(seqA, seqB, n, m-1) $+ \sum_{k=1}^{n} BP(k,m) \cdot NAB(seqA, seqB, k-1, m-1) \cdot N1(seqA, k+1, n) + \sum_{k=1}^{m-\Theta-1} BP(k,m) \cdot NAB(seqA, seqB, n, k-1) \cdot N1(seqB, k+1, m-1)$
10: **sinon si** $n > \Theta + 1$ **et** $m \leq \Theta + 1$ **alors**
11: **retourner** NAB(seqA, seqB, n-1, m) $+ \sum_{k=1}^{m} BP(n,k) \cdot NAB(seqA, seqB, n-1, k-1) \cdot N1(seqB, k+1, m) + \sum_{k=1}^{n-\Theta-1} BP(k,n) \cdot NAB(seqA, seqB, k-1, m) \cdot N1(seqA, k+1, n-1)$
12: **sinon**
13: **retourner** BP(n, m) · NAB(seqA, seqB, n-1, m-1) + NAB(seqA, seqB, n-1, m-1) $+ \sum_{k=1}^{m-1} BP(n,k) \cdot NAB(seqA, seqB, n-1, k-1) \cdot N1(seqB, k+1, m) + \sum_{k=1}^{n-\Theta-1} BP(k,n) \cdot NAB(seqA, seqB, k-1, m-1) \cdot N1(seqA, k+1, n-1) + \sum_{k=1}^{n-1} BP(k,m) \cdot NAB(seqA, seqB, k-1, m-1) \cdot N1(seqB, k+1, n) + \sum_{k=1}^{m-\Theta-1} BP(k,m) \cdot NAB(seqA, seqB, n-1, k-1) \cdot N1(seqB, k+1, m-1)$ + N1(seqA, 1, n-1) · N1(seqB, 1, m-1)
14: **fin si**
15: **fin Procédure**

Ici, les algorithmes 6 et 7 sont présentés de façon récursive. En pratique, nous les avons implémenté en type de programmation dynamique. Les valeurs retournées par les fonctions $N1(seq, i, j)$ et $NAB(seqA, seqB, i, j)$ sont en fait stockées dans deux tableaux à deux dimensions.

2. Pour la température $T \in \{0°C, ..., 100°C\}$, utiliser notre programme `RNA-WL`, pour calculer les densités relatives d'énergie : $g(A,T), g(B,T), g(AA,T), g(BB,T)$ et $g(AB,T)$

3. Pour la température $T \in \{0°C, ..., 100°C\}$, calculer les fonctions de partition des cinq espèces : $Z(A,T), Z(B,T), Z(AA,T), Z(BB,T)$ et $Z(AB,T)$:

4. Pour la température $T \in \{0°C, ..., 100°C\}$, calculer l'énergie libre d'ensemble : $\Delta G(A,T), \Delta G(B,T), \Delta G(AA,T), \Delta G(BB,T)$ et $\Delta G(AB,T)$ en utilisant la méthode de Dimitrov et Zuker [67]. Pour cela, nous effectuons les étapes de a à f.

   a Effectuer les corrections de redondance :

$$\begin{aligned} Z(AA,T) &= Z(AA,T) - Z(A,T)^2 \\ Z(BB,T) &= Z(BB,T) - Z(B,T)^2 \\ Z(AB,T) &= Z(AB,T) - Z(A,T) \cdot Z(B,T) \end{aligned} \tag{3.24}$$

   b Effectuer les corrections de symétrie :

$$\begin{aligned} Z(AA,T) &= \frac{Z(AA,T)}{2} \\ Z(BB,T) &= \frac{Z(BB,T)}{2} \end{aligned} \tag{3.25}$$

   c Calculer les constantes d'équilibre dans la réaction chimique dépandante de la température :

$$\begin{aligned} K_A &= \frac{Z(AA,T)}{Z(A,T)^2} \\ K_B &= \frac{Z(BB,T)}{Z(B,T)^2} \\ K_{AB} &= \frac{Z(AB,T)}{Z(A,T) \cdot Z(B,T)} \end{aligned} \tag{3.26}$$

d Calculer les concentrations des molécules A et B en l'état d'équilibre :

$$\begin{aligned} 2 \cdot K_A \cdot N_A^2 + K_{AB} \cdot N_A \cdot N_B + N_A - N_A^0 = 0 \\ 2 \cdot K_B \cdot N_B^2 + K_{AB} \cdot N_A \cdot N_B + N_B - N_B^0 = 0 \end{aligned} \tag{3.27}$$

où les concentrations des molécules A et B en l'état initial N_A^0, N_B^0 sont supposées comme les valeurs connues et les constantes d'équilibre K_A, K_B, K_{AB} sont déjà obtenues dans l'étape précédente. Pour savoir les concentrations des molécules A et B en l'état d'équilibre N_A et N_B, nous pouvons par example utiliser la méthode de Newton pour résoudre ces deux équations non linéaire. En raison du problème d'instabilité numérique, nous avons choisit la méthode de la recherche binaire expliquée dans la thèse de Nicholas R. Markham(p. 43 of [68]).

e Calculer l'énergie libre d'ensemble ΔG :

$$\begin{aligned} \mu_A &= -RT \ln(Z_A) + RT \ln\left(\frac{N_A}{N_A^0}\right) \\ \mu_B &= -RT \ln(Z_B) + RT \ln\left(\frac{N_B}{N_B^0}\right) \\ \mu_{AB} &= -RT \ln(Z_{AB}) + RT \ln\left(\frac{N_{AB}}{N_A^0 \cdot N_B^0}\right) \\ \mu_{AA} &= -RT \ln(Z_{AA}) + RT \ln\left(\frac{N_{AA}}{N_A^0 \cdot N_A^0}\right) \\ \mu_{BB} &= -RT \ln(Z_{BB}) + RT \ln\left(\frac{N_{BB}}{N_B^0 \cdot N_B^0}\right). \end{aligned} \tag{3.28}$$

L'énergie libre d'ensemble satisfait :

$$\Delta G = \mu_A \cdot N_A + \mu_B \cdot N_B + \mu_{AA} \cdot N_{AA} + \mu_{BB} \cdot N_{BB} + \mu_{AB} \cdot N_{AB} \tag{3.29}$$

qui peut être simplifiée par

$$\Delta G = \mu_A \cdot N_A^0 + \mu_B \cdot N_B^0 \tag{3.30}$$

f Normaliser l'énergie libre d'ensemble en termes de l'énergie par mole dans la solution :

$$\Delta G = \frac{\Delta G}{max(N_A^0, N_B^0)} \tag{3.31}$$

5. Les quantités thermodynamiques peuvent être déterminées comme suit :

   l'entropie :

$$\triangle S = -\frac{\partial \Delta G}{\partial T} \tag{3.32}$$

   l'enthalpie :

$$\triangle H = \triangle G - T \cdot \frac{\partial \Delta G}{\partial T} \tag{3.33}$$

   la capacité de chaleur :

$$C_p(T) = \frac{\partial \Delta H}{\partial T} = -T\frac{\partial^2 \Delta G}{\partial T^2} \tag{3.34}$$

   L'approximation de la dérivée seconde de $\triangle G$ peut être calculée par l'expression suivante en utilisant les $(2m+1)$ points autour de la température T [68] :

$$\frac{\partial^2 \Delta G}{\partial T^2} \approx \frac{30}{m(m+1)4m^2(2m+3)\delta T^2}\Sigma_{-m\leq i\leq m}(3i^2 - m(m+1)\Delta G(T+i\delta T) \tag{3.35}$$

   où δT est l'écart de la température T, ici cette valeur est 1 degré Kelvin.

6. Dans la dernière étape, nous allons lisser la courbe de la capacité de chaleur avec une fenêtre glissante, la valeur est remplacée par la moyenne des 11 valeurs autour. La capacité de chaleur est intéressante, parce que son maximum local indique la température de dénaturation.

3.5.3.2 Prédiction de la température de dénaturation.

La figure 3.12 illustre une comparaison de la capacité de chaleur obtenue par notre pipeline et celle du programme `UNAfold`. Les maxima locaux de ces deux courbes indiquent les températures de dénaturation, qui sont proches.

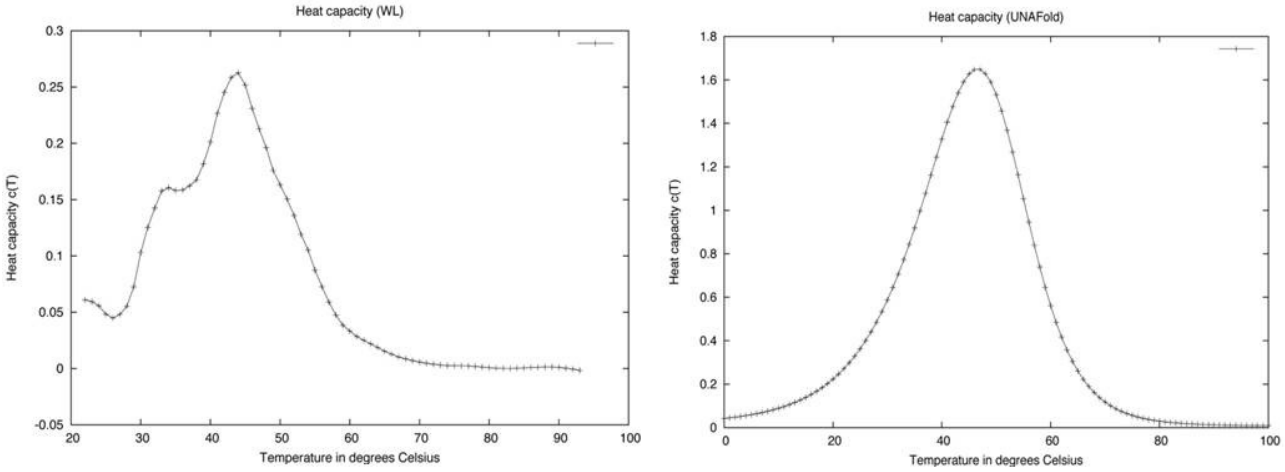


FIGURE 3.12 – Les courbes de la capacité de chaleur sur une petite séquence 5′-`AGCGA`-3′, hybridé avec son complément inverse 3′-`UCGCU`-5′. La figure de gauche est générée par `RNA-WL`. La figure de droite est générée par le programme `UNAFOLD` [12].

Séquence	Expériment	UNAFold	RNAcofold	`RNA-WL`
ACGCA&UGCGU	29,8	42,64	46,14	**42**
GCACG&CGUGC	37,5	46,61	**43,91**	44
AGCGA&UCGCU	30,2	42,68	45,15	**41**
GCUCG&CGAGC	37,2	47,75	**44,71**	48
ACUGUCA&UGACAGU	48,2	56,8	57,59	**51**
GUCACUG&CAUGUAC	51,1	58,44	**55,91**	56
AGUCUGA&UCAGACU	45,7	56,4	56,68	**52**
GACUCAG&CUGAGUC	52	59,11	56,25	**52**
GAGUGAG&CUCACUC	53,7	59,07	**56,00**	58

TABLE 3.4 – Les températures de dénaturation expérimentales et celles prédites par le programme UNAFold, RNAcofold et notre pipeline. Les données experimentales sont extraites de l'article [35]. Les données du programme `UNAfold` et `RNAcofold` sont extraites de l'article [41]

La table 3.4 présente les températures de dénaturation prédites par notre pipeline, par `UNAFold` et par `RNAcofold`. Comme on l'a déjà vu, parmi les 9 séquences testées, il y a 5 séquences sur lesquelles notre pipeline a predit la températures de dénaturation la plus proche

de la valeur expérimentale. La performance de notre pipeline est meilleure que celle des deux autres programmes.

3.6 Discussion

Dans ce chapitre, nous avons présenté un nouvel algorithme `RNA-WL` qui permet de prédire la densité relative d'états d'énergie des structures secondaires d'une séquence ou d'une hybridation de deux séquences d'ARN. Nous avons calculé le nombre de structures et d'hybridations séparément, nous obtenons la densité absolue d'états d'énergie ce qui donne ensuite la fonction de partition et la température de dénaturation dans le cas de l'hybridation. Notre programme `RNA-WL` prédit beaucoup plus rapidement la densité d'états d'énergie que le programme `RNAsubopt`. Pour la plupart des séquences testées, notre pipeline prédit mieux la température de dénaturation que les deux autres programmes existants. Cependant, le vrai avantage de notre algorithme `RNA-WL` est qu'il n'y a pas de restriction sur les interactions autorisées. Contrairement aux approches de programmation dynamique, toutes les structures secondaires et hybridations peuvent être générées par notre échantillonnage. Si nous avons un modèle d'énergie pour les structures secondaires avec pseudo-noeuds, le problème NP-complet de la prédiction des quantités thermodynamiques des structures secondaires avec pseudo-noeuds pourrait être résolu de façon approchée par notre algorithme `RNA-WL`.

Chapitre 4

Les structures MEA [1] à différentes distances de paires de bases d'une structure secondaire d'ARN

4.1 Introduction

Au cours des dernières années, plusieurs découvertes ont montré que la transcription de gènes essentiels chez les bactéries et les eucaryotes est souvent contrôlée par les structures de certaines molécules d'ARN. Les *riboswitchs* (riborégulateurs) sont des éléments de contrôle qui reconnaissent directement un métabolite cellulaire et qui régulent des gènes localisés en aval.

Les riboswitchs régulent les gènes en effectuant un changement allostérique, c'est-à-dire une commutation entre deux structures distinctes, appelées les structures du "gène on"/"gène off" ou les structures fonctionnelles de riboswitch. Il est donc important de développer un algorithme pour les prédire. Certains outils, tels que `paRNAss` [69], `RNAshapes` [70], `RNAbor` [71], peuvent être utilisés pour prédire les structures fonctionnelles de riboswitch. Néanmoins, les outils existants ne peuvent pas prédire avec précision toutes les structures fonctionnelles des riboswitchs. Ils se cantonnent à un type de famille spécifique, ou bien ne prédisent correctement que les parties bien conservées des structures. Par conséquence, le développement des algorithmes supplémentaires

1. MEA : "Maximum Expected Accuracy" en anglais.

pour la prédiction des structures fonctionnelles de riboswitch est important.

D'autre part, nous avons observé que la différence des énergies libres des deux structures fonctionnelles de riboswitch peut être aussi grande que 15-20 kcal/mol, du coup, les algorithmes basés sur la minimisation d'énergie libre ne conviennent plus à la prédiction des structures fonctionnelles de riboswitch. De plus, les études dans [72] montrent que les structures secondaires d'ARN sont plus précisément prédites par les structures MEA que par les structures MFE. La structure MEA s a le score EA maximum, le score EA est calculé par la formule suivante :

$$EA(s) = 2 \cdot \sum_{(i,j) \in s} p_{i,j} + \sum_{i \; est \; non-appariée} q_i$$

où la première somme est sur toutes les paires de bases appariées dans la structure s, la deuxième somme est sur toutes les bases non-appariées dans s, $p_{i,j}$ [resp. q_i] est la probabilité que les bases i, j sont appariées [resp. la base i est non-appariée] dans l'ensemble des structures à basse énergie. Donc, le développement d'un algorithme basé sur les structures MEA semble une idée raisonnable pour prédire les structures fonctionnelles de riboswitch.

Je m'intéresse à la recherche d'un algorithme qui engendre des structures sous-optimales, dans lesquelles, nous espérons trouver deux structures fonctionnelles de riboswitch. Il existe certaines méthodes de génération des structures sous-optimales d'ARN, comme Zuker *et al.* [73], Ding *et al.* [74], Wuchty *et al.* [75].

J'ai donc développé et implémenté un nouvel algorithme `RNAborMEA` qui produit des structures sous-optimales à partir d'une structure initiale où les structures fonctionnelles sont susceptible de se trouver. Je l'ai appliqué sur une séquence de riboswitch TPP et sur l'ensemble des séquences de la famille de riboswitch purine.

4.2 Riboswitch

Les riboswitchs sont des domaines d'ARNm qui reçoivent des ligands spécifiques. Ces domaines se trouvent dans les parties non codantes, principalement dans le 5' UTR des procaryotes. Ils contrôlent les expressions des gènes par des changements allostériques qui sont provoqués par les liaisons avec les ligands.

Les riboswitchs sont souvent conceptuellement divisés en deux parties : un aptamère qui est capable de fixer un ligand spécifique, dont la séquence et la structure sont très conservées dans différentes espèces et une plateforme d'expression qui subit des changements structurels en réponse à l'évolution de l'aptamère. Les séquences de la plateforme d'expression sont peu conservées.

Le phénomène du changement allostérique joue un rôle essentiel dans un certain nombre de processus biologiques, comme la régulation de la réplication virale [77] et de la réplication du viroïde [78], la régulation de la transcription et de la traduction chez les procaryotes [4], la régulation de l'épissage alternatif chez les eucaryotes [5], la régulation des gènes de réponse au stress chez les humains [79], *etc.*

Bien qu'ils existent aussi chez certaines plantes et champignons, les riboswitchs ont été principalement observés chez les bactéries. Les études récentes [80] ont distingué sept classes d'aptamères naturels de riboswitchs qui reconnaissent huit métabolites :

- riboswitch coenzyme-B_{12} [81, 82, 83].
- riboswitch thiamine pyrophosphate (TPP) [84, 85, 86, 87].
- riboswitch flavin mononucleotide (FMN) [88].
- riboswitch guanine/adenine (purine) [4, 89].
- riboswitch S-adenosylmethionine (SAM) [90, 91, 92].
- riboswitch lysine [93, 94].
- riboswitch Glucosamine-6-phosphate (GlcN6P) [95, 96].

Chaque classe de riboswitch a un consensus de motifs de séquence et de structure dans la région de l'aptamère, les différents repliements des aptamères sont nécessaires pour former les sites de liaison des ligands spécifiques. Une exception réside dans la classe de riboswitch purine, il a été montré que son aptamère a d'abord la spécificité de se lier avec la guanine, mais il peut échanger sa spécificité moléculaire pour l'adénine par une mutation ponctuelle unique [4, 89].

Dans [80], Mandal et Breaker ont identifié les deux structures fonctionnelles de la famille de riboswitch Coenzyme-B_{12}, ils ont également expliqué leurs mécanismes de régulation de la transcription et de l'initiation de la traduction. La figure 4.1 extraite de [80] montre que la transcription d'ARN est contrôlée par la formation d'une tige anti-terminatrice dans la structure du "gène on" [2] (voir la figure 4.1 à gauche) ou d'une tige terminatrice intrinsèque dans

2. la structure du "gène on" : la structure qui permet de completer la transcription et l'expression du gène.

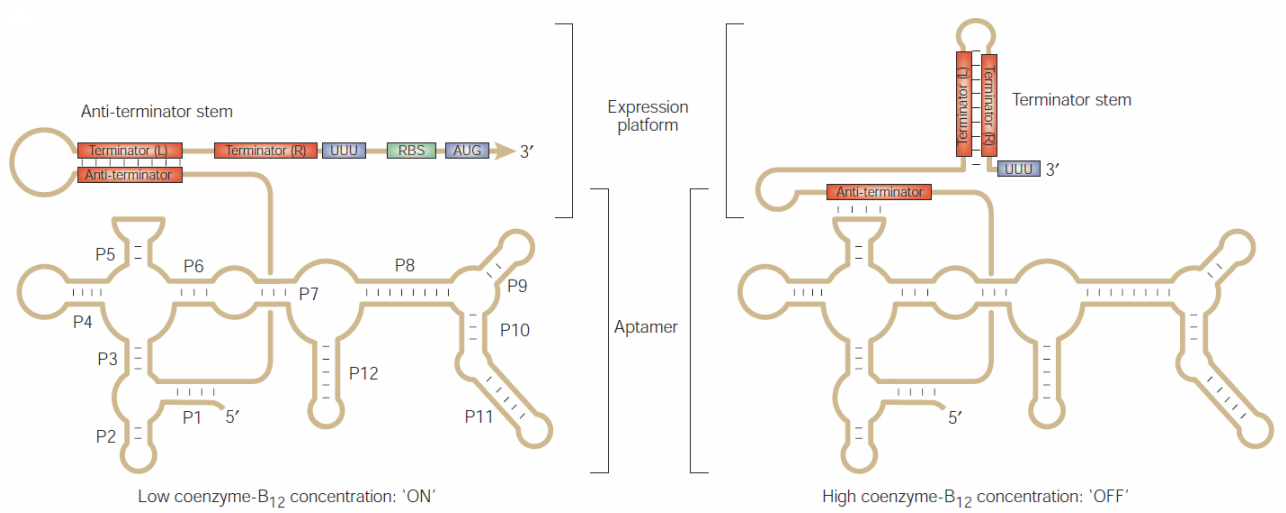


FIGURE 4.1 – Les deux structures fonctionnelles du riboswitch *Coenzyme* – B_{12} et leurs mécanismes de contrôle de la transcription d'ADN [80].

la structure du "gène off" [3] (voir la figure 4.1 à droite). Lorsque la quantité de la Coenzyme-B_{12} est faible, la transciption produit des ARNm dans lesquels l'aptamère reste non lié avec le ligand, une tige anti-terminatrice est formée, ce qui va empêcher la formation de la tige terminatrice et permettre de completer la transcription. Cependant, lorsque la quantité de la Coenzyme-B_{12} est importante, le ligand (la Coenzyme-B_{12}) va se lier avec l'aptamère. Cela implique que les bases de la boucle $P5$ vont se lier avec des bases dans la zone anti-terminatrice, la tige terminatrice va être formée. Du coup, la transcription et l'expression du gène est empêchée.

En raison de l'importance biologique des riboswitchs, plusieurs groupes ont développé des algorithmes qui tentent de reconnaître les riboswitchs, parmi lesquels il y a certaines méthodes basée sur la thermodynamique, comme les programmes `RNAbor` [71], `RNAshapes` [97], et `paRNAss` [70]. D'autres parts, en raison de la conservation de la séquence et de la structure au sein de l'aptamère, les algorithmes existants, comme [98, 99, 100], tentent de détecter seulement les aptamères des riboswitchs sans les plateformes d'expression. En plus de ces algorithmes, les autres outils qui s'appuient sur des grammaires non contextuelles probabilistes, comme `Infernal` [7] et `CMFinder` [101], peuvent être utilisés pour reconnaître les aptamères des riboswitchs. En particulier, `Infernal` est utilisé pour créer la base de données Rfam [112] qui contient actuellement 22 familles de aptamères de riboswitchs parmi les familles d'ARN non codant repertoriées.

3. la structure du "gène off" : la structure qui empêcher la transcription et l'expression du gène.

4.3 Le programme `RNAbor`

Freyhult *et al.*[76] ont proposé un nouvel outil, `RNAbor`, afin d'étudier les structures secondaires sous-optimales d'ARN. Cet outil calcule les statistiques concernant les structures k-voisines d'une séquence et d'une structure donnée. Pour chaque distance de paires de bases k, il calcule le nombre, la densité, la probabilité de Boltzmann exacte des structures k-voisines et les structures $MFE(k)$. Ils ont définit la distance de paires de bases, les structures k-voisines, et les structures $MFE(k)$ comme suit :

Soit w une séquence d'ARN, et s_0, s_1, s_2 trois structures secondaires possibles pour w, $L_{bp}(s_1)$ l'ensemble des paires de bases de la structure secondaire s_1.

Définition 4.1 ***La distance de paires de bases*** *entre deux structures s_1 et s_2, notée $D_{bp}(s_1, s_2)$, est le nombre de paires de bases dans $L_{bp}(s_1)$ et pas dans $L_{bp}(s_2)$ plus le nombre de paires de bases dans $L_{bp}(s_2)$ et pas dans $L_{bp}(s_1)$:*

$$D_{bp}(s_1, s_2) = Card(L_{bp}(s_1) \backslash L_{bp}(s_2)) + Card(L_{bp}(s_2) \backslash L_{bp}(s_1))$$

Définition 4.2 ***Les structures k-voisines de*** *s_0 sont les structures de w à la distance de paires de bases k de la structure s_0.*

Nous rappelons que la structure MFE de w est la structure ayant l'énergie libre minimume parmi toutes les structures de w.

Définition 4.3 ***Les structures*** *$MFE(k)$ de la séquence w sont les structures ayant l'énergie libre minimum parmi toutes les structures k-voisines de s_0.*

Freyhult *et al.* [76] ont appliqué le programme `RNAbor` à la recherche des structures fonctionnelles du riboswitch SAM. Ce riboswitch se trouve dans un certain nombre de gènes qui codent pour des protéines impliquées dans la biosynthèse de la méthionine ou de la cystéine chez les bactéries Gram positives.

Ils calculent les probabilités de Boltzmann des structures k-voisines en fonction de la distance de paires de bases pour une séquence de riboswitch SAM (voir la figure 4.2). Ils ont observé qu'il y a un pic à la distance 30, ce qui signifie que les structures 30-voisines ont une probabilité de Boltzmann plus grande que les autres structures. Du coup, ils comparent la structure MFE

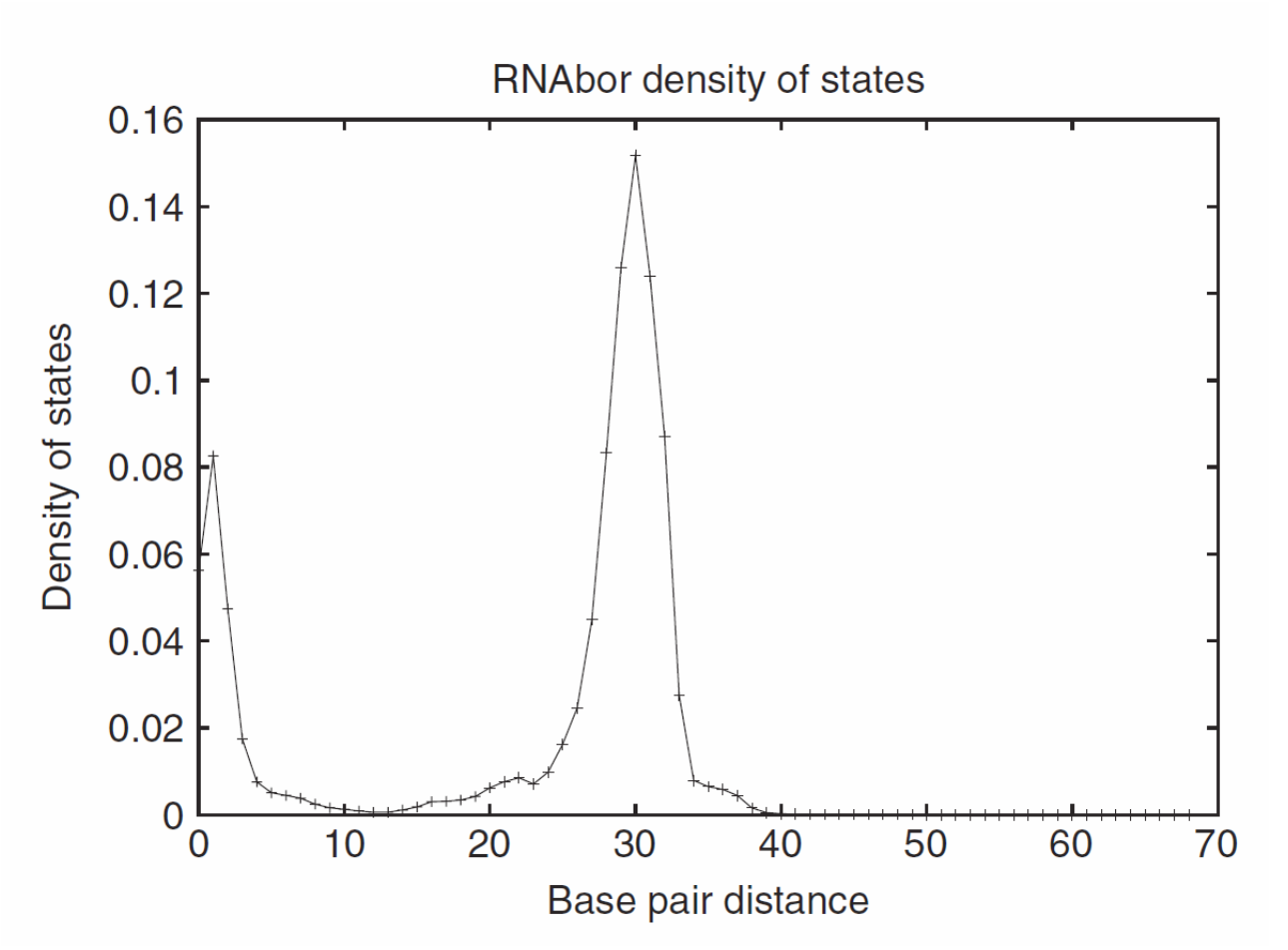


FIGURE 4.2 – La probabilité de Boltzmann des structures k-voisines du riboswitch SAM en fonction de la distance de paires de bases.[76]

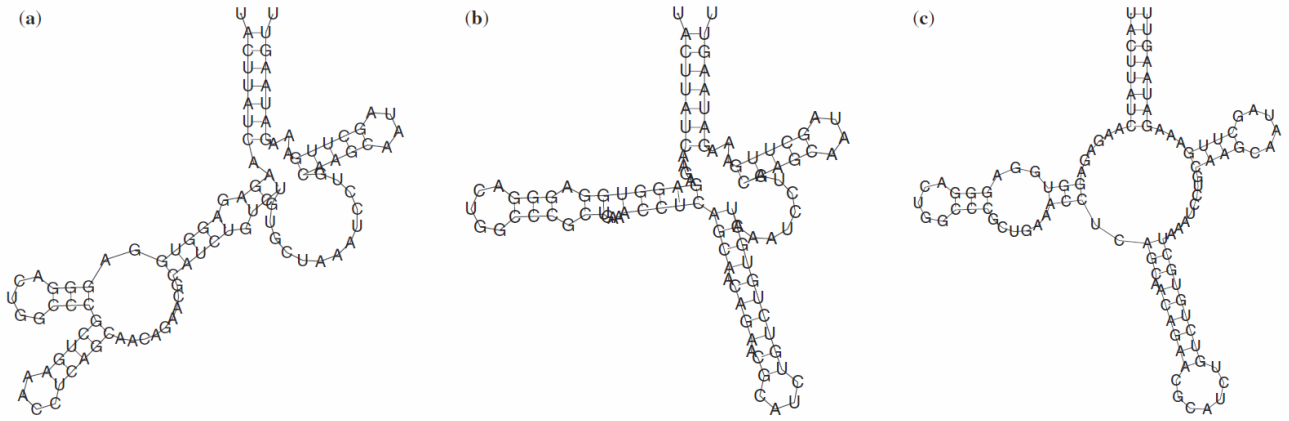


FIGURE 4.3 – La structure MFE (a), la structure MFE(30) (b), et la structure réelle (c) du riboswitch SAM.[76]

et la structure MFE(30) avec la structure réelle du riboswitch SAM, dont l'identifiant EMBL est AP004597.118941-119041 (voir la figure 4.3). Ils remarquent que la structure MFE(30) est clairement beaucoup plus proche de la structure réelle que la structure MFE. Les figures 4.2 et 4.3 ont donc montré que le programme `RNAbor` donne une bonne compréhension des aspects structurels du riboswitch SAM.

4.4 La structure secondaire MEA

Do *et al.* [103] ont proposé une méthode alternative pour chercher la structure la plus "probable", nous l'appelons la structure secondaire MEA. Cette structure est déterminée en maximisant le score EA qui est la somme des probabilités des paires de bases et des probabilités des nucléotides non-appariés d'une structure d'ARN.

Soit w une séquence d'ARN, s une structure secondaire possible de w.

Définition 4.4 ***Le score EA*** *de la structure s est défini comme suit :*

$$EA(s) = \sum_{(i,j)\in s} 2 \cdot \alpha \cdot p(i,j) + \sum_{i\ est\ non-apparieé} \beta \cdot q_i \tag{4.1}$$

où la première somme est calculée sur toutes les positions des paires de bases dans la structure s, et la deuxième somme est calculée sur toutes les positions des bases non-appariées. α et β sont les paramètres prédéfinis par les utilisateurs.

Définition 4.5 ***La structure MEA de*** *w est l'ensemble des structures ayant le score EA maximum parmi toutes les structures de w.*

Pour prédire les structures MEA, Do *et al.* [103] proposent d'itérer les étapes suivantes :

1. calculer les probabilités $p(i,j)$ que les nucléotides aux positions i et j soient appariés en utilisant une grammaire non-contextuelle stochastique.

2. calculer les probabilités que le nucléotide à la position i soit non-apparié :

$$q_i = 1 - \sum_{j \neq i} p(i,j). \tag{4.2}$$

3. utiliser un algorithme de type programmation dynamique, qui est similaire à celui de Nussinov et Jacobson [104] pour déterminer la structure MEA de la séquence w.

Subséquemment, Kiryu *et al.* [106] calculent la structure MEA en remplaçant la grammaire non-contextuelle stochastique par l'algorithme de McCaskill [107] qui calcule la probabilité de Boltzmann d'appariement de paire de bases en utilisant la formule suivante :

$$p(i,j) = \frac{\sum_{(i,j) \in s} e^{-E(s)/RT}}{\sum_s e^{-E(s)/RT}} \tag{4.3}$$

où $E(s)$ est l'énergie libre de la structure s, à l'égard du modèle de Turner [35], T est la température absolue en Kelvin, R est la constante universelle des gaz parfaits.

Par conséquent, $p(i,j)$ est la somme des facteurs de Boltzmann de toutes les structures secondaires contenant la paire de bases (i,j), divisée par la fonction de partition de Boltzmann, c'est-à-dire la somme des facteurs de Boltzmann de toutes les structures secondaires.

4.5 Méthode de `RNAborMEA`

Nous avons développé un algorithme `RNAborMEA` qui est similaire à l'algorithme `RNAbor`. Toutefois, au lieu de chercher les structures ayant l'énergie libre minimume parmi les structures k-voisines, nous cherchons les structures ayant le score EA maximum parmi les structures k-voisines. Notre programme `RNAborMEA` propose un ensemble de structures sous-optimales et suggère les structures fonctionnelles de riboswitch. Ce travail est motivé par les raisons suivantes : 1) Les algorithmes de la détection de riboswitchs indiqués dans la section 4.2 sont tous basés sur la séquence et la structure conservées dans la région de l'aptamère, ils ne prédisent

pas la location de la plateforme d'expression ni les deux structures fonctionnelles. 2) Le programme `RNAbor` nous montre une voie possible pour chercher les structures fonctionnelles de riboswitchs parmi les structures k-voisines. 3) La différence d'énergie libre entre deux structures fonctionnelles de riboswitch peut être aussi grande que 15-20 kcal/mol, donc les méthodes qui prédisent les structures MFE ne conviennent plus à la prédiction des structures fonctionnelles de riboswitchs. D'autre part, intuitivement, le score EA semble être une mesure de la précision du repliement attendu, car il a été précédemment démontré que les paires de bases avec une forte probabilité d'appariement sont plus susceptibles d'être dans la structure reconnue [105]. De plus, les études de Mathews *et al.* [72] montrent que les structures MEA sont en moyennne plus proches des structures natives que les structures MFE.

4.5.1 Les structures $MEA(k)$

Nous rappelons que pour la séquence w et pour une structure s_0 de w, la structure MEA est la structure ayant le score EA maximum parmi l'ensemble des structures possibles de w.

Soit w une séquence d'ARN, s_0 une structure possible de w, k un nombre entier.

Définition 4.6 ***Les structures*** $MEA(k)$ *sont les structures ayant le score EA maximum parmi toutes les structures k-voisines de s_0.*

4.5.2 Méthode

Étant donnée une séquence w de taille n, une structure s_0 de w, un nombre entier K_{max}, notre programme `RNAborMEA` calcule les scores EA pour toutes les sous-séquences de w et pour toutes les distances de paires de bases de s_0. Nous stockons ces scores dans un tableau M à 3 dimensions de taille $n \times n \times (K_{max} + 1)$, où la valeur de $M(i, j, k)$ est le score EA maximum de l'ensemble des structures qui correspondent à la sous séquence $w(i, j)$ et qui sont à la distance de paires de bases k de s_0. Au final, notre programme `RNAborMEA` propose les structures $MEA(k)$ dont les scores EA sont stockés dans les cases $M(1, n, k)$.

Notre programme `RNAborMEA` calcule les valeurs de $M(i, j, k)$ en utilisant une récurrence sur l'augmentation des valeurs de $(j - i)$ et des valeurs de k. C'est-à-dire que sauf l'initialisation

de certaines valeurs dans le tableau M, toutes les autres valeurs $M(i, j, k)$ vont être calculées à partir les valeurs $M(i', j', k')$ déjà connues (voir l'algorithme 8).

Pour trouver les structures $MEA(k)$ de la séquence w et de la structure s_0, nous itérons les 4 étapes suivants :

1. Initialiser un tableau à 3 dimensions M de taille $n \times n \times K_{max}$.

2. Pour toutes les positions i, j, telles que $1 \leq i \leq j \leq n$, calculer les probabilités $p(i, j)$ que les nucléotides aux positions i et j soient appariées (voir la formule 4.3) et les probabilités q_i que le nucléotide à la position i soit non-apparié (voir la formule 4.2) en utilisant l'algorithme de McCaskill [107]. Pour cela, nous avons utilisé le programmme `RNAfold` avec l'option -p [38].

3. Utiliser nos algorithmes 8 et 9 de type programmation dynamique pour calculer toutes les valeurs de $M(i, j, k)$ par les décompositions de la séquence. Quand la taille de la sous-séquence $w(i, j)$ est supérieure à $(\theta + 1)$ [4], alors le score EA maximum des structures de la séquence w qui sont à la distance de paires de bases k de s_0 est la valeur maximale parmi celles calculées selon les points (a), (b) et (c) suivants :

   (a) Quand le nucléotide à la position j est non-apparié, la séquence $w(i, j)$ peut être traitée en 2 parties : la séquence $w(i, j-1)$ et le nucléotide j.

   Soit $b_0 = 1$, si le nucléotide j est apparié avec un autre nuclétide dans la structure s_0, soit $b_0 = 0$, sinon.
   La valeur de $M(i, j, k)$ est calculée par la formule suivante :

$$M(i, j, k) = M(i, j-1, k-b_0) + \beta \cdot q_j \tag{4.4}$$

   (b) Quand le nucléotide à la position j est apparié avec le nucléotide à la position i, la séquence $w(i, j)$ peut être traitée en 2 parties : la séquence $w(i+1, j-1)$ et la paire de bases (i, j).

4. θ est le nombre minimum des nucléotides entre une paire de bases, par défaut, $\theta = 3$

Soit $b_1 = 0$, si la paire de bases (i,j) appartient à s_0, soit $b_1 = 1$, sinon.
La valeur de $M(i, j, k)$ est calculé par la formule suivante :

$$M(i, j, k) = M(i, j - 1, k - b_1) + \alpha \cdot p(i, j) \tag{4.5}$$

(c) Quand le nucléotide à la position j est apparié avec le nucléotide à la position r, où $i < r < j$.

Nous cherchons le score EA maximum parmi toutes les décompositions de la séquence $w(i, j)$ selon le nucléotide r et parmi toutes les combinaisons des distances de paires de bases. D'abord, La séquence $w(i, j)$ peut être traitée en 3 parties : les séquences $w(i+1, r-1)$, $w(r+1, j-1)$ et la paire de bases (r, j). Ensuite, la distance de paire de bases est aussi divisée en 3 parties : k_0, k_1, et b_2, où $k_0 + k_1 + b_2 = k$.

- Soit k_0 la distance de paires de bases entre les structures de la sous séquence $w(i, r - 1)$ et la structure $s_0(i, r - 1)$.
- Soit k_1 la distance de paires de base entre les structures de la sous séquence $w(r + 1, j - 1)$ et la structure $s_0(r - 1, j - 1)$.
- Soit b_2 le nombre de paires de bases dans s_0 dont le premier nucléotide appartient à la zone $[i, r - 1]$ et le deuxième nucléotide appartient à la zone $[r + 1, j - 1]$.

La valeur de $M(i, j, k)$ est calculée par la formule suivante :

$$M(i, j, k) = \max_{i<r<j,\ 0\leq k_0\leq k-b_2,\ k_1=k-k_0-b_2} \{M(i, r-1, k_0)+2\cdot\alpha\cdot p_{r,j}+M(r+1, j-1, k_1)\ \} \tag{4.6}$$

4. Au moment de calculer la valeur de $M(i, j, k)$, nous mémorisons en même temps les

nombres (r, k_0, k_1) dans la case $M(j, i, k)$, ce que nous permet de retrouver les structures $MEA(k)$ en "back-track" plus tard.

De plus, nous avons étendu le programme `RNAborMEA` pour tenir compte de contraintes structurelles. Pour une séquence donnée, ses contraintes structurelles ont la même taille que la séquence, elles sont composées par les symboles "(", ")", "*", ".". Nous prenons un exemple de la séquence "GGAAACCU", dont une contrainte structurelle possible est "((***)).". Dans toutes les structures qui tiennent compte de cette contraite, le nucléotide à la position 1 [resp. 2] doit être apparié avec celui à la position 7 [resp. 6], les nucléotides aux positions 3, 4 et 5 doivent être non-appariés, le nucléotide à la position 8 peut être apparié ou non-apparié.

Distance	EA score	Structure MEA(k)	Energy
0	46.116787	(((((((.(((...(((.....))).))).)))))))	-20.53
1	44.906365	(((((((.(((...(((....(.....)....))).))).)))))))	-17.5
2	43.6942	(((((((.(((...(((....((.....))....))).))).)))))))	-19.1
3	42.436394	(((((((.(((...((.....).).).))).)))))))	-12.87
4	41.665097	(((((((.(((...(.(.....(....))).))).)))))))	-8.3
5	40.621324	(((((((.(((...((.....))..))).)))))))	-18.17
6	39.754057	(((((((.(((....((.....(....))).))).)))))))	-11.5
7	38.585896	(((((((.(((...((.....)))..)).)))))))	-8.87
8	37.750371	(((((((.((....(((.....(....))).))).)))))))	-6.2
9	36.538206	(((((((.((....(((.....(.....)(....))).))).)))))))	-3.1
10	35.183078	(((((((.((....(((..(.((.....))....))).))).)))))))	-4.9
11	33.782144	(((((((.((....)((.....(....))).)...)))))))	2.6
12	32.56998	(((((((.((....)((.....(.....)(....))).)...)))))))	5.7
13	31.214852	(((((((.((....)((..(.((.....))....))).)...)))))))	3.9
14	29.726035	(((((((.((....)((..(.(((...)))....))).)...)))))))	3.7

FIGURE 4.4 – L'illustration d'une partie de la sortie du programme `RNAborMEA` avec contrainte structuelle.

Les figures 4.4 et 4.5 illustrent les sorties du programme `RNAborMEA` avec ou sans la contrainte structurelle "(((((((*.....*))))))" pour la séquence "CGCCACC-CUGCGAACCCAAUAAUAAAAUAUACAAGGGAGCAAGGUGGCG" et pour la structure initiale "(((((((.(((...(((.....))).))).))))))".

Dans la figure 4.4 [resp. 4.5], la première colonne indique les distances de paires de bases de la structure initiale, la deuxième colonne indique les scores EA maximums parmi les structures k-voisines en tenant compte de la contrainte structurelle [resp. sans tenir compte de la

Distance	MEA	Structure	Energy
0	46.116787	(((((((.(((...(((.....))).))).)))))))	-20.53
1	45.48998	.((((((.(((...(((.....))).))).)))))).	-20.13
2	44.279557	.((((((.(((...(((....(.....)....))).))).)))))).	-17.1
3	43.067392	.((((((.(((...(((....((.....))....))).))).)))))).	-18.7
4	41.809587	.((((((.(((...((.....).).).))).)))))).	-12.47
5	41.03829	.((((((.(((...(.(.....(....))).))).)))))).	-7.9
6	39.994516	.((((((.(((...((.....))..))).)))))).	-17.77
7	39.12725	.((((((.(((....((.....(....))).))).)))))).	-11.1
8	37.959088	.((((((.(((...((.....)))..)).)))))).	-8.47
9	37.123563	.((((((.((....(((.....(....))).))).)))))).	-5.8
10	35.911398	.((((((.((....(((.....(.....)(....))).))).)))))).	-2.7
11	34.556271	.((((((.((....(((..(.((.....))....))).))).)))))).	-4.5
12	33.175232	.((((.(.(((....((.....(....))).))..)))))).	2
13	32.15043	.((((.((((....(((.....(....))).))).)))))).	-1.4
14	31.171545	.((((.(.((....(((.....(....))).))..)))))).	7.3
15	29.959381	.((((.(.((....(((.....(.....)(....))).))..)))))).	10.4
16	28.725524	.((((((.((....)(.....(.....)....))).)(...))))).	21
17	27.51336	.((((((.((....)(.....((.....))....))).)(...))))).	19.4
18	26.375613	.((((...)((...(((.....(.(....))).)...)))))).	21.3
19	25.179047	.((((...))(....((.....(....))).((...))))).	15.9
20	24.173487	.((((...))((..(((.....(....))).)(...))))).	19.3
21	23.173244	.((.(...))(...(((.....(....))).((...))))).	28.3
22	21.961079	.((.(...))(...(((.....(.....)(....))).((...))))).	31.4
23	20.834128	.(...)(..((...(((.....(....))).((...))))))	22.6
24	19.830126	(((((...))((..(((.....(....))).))..))).)..	21.7
25	18.617961	(((((...))((..(((.....(.....)(....))).))..))).)..	24.8
26	17.262834	(((((...))((..(((..(.((.....))....))).))..))).)..	23
27	15.774017	(((((...))((..(((..(.(((...)))....))).))..))).)..	22.8

FIGURE 4.5 – L'illustration d'une partie de la sortie du programme `RNAborMEA` sans contrainte structurelle.

contrainte structurelle], la troisième colonne contient une des structures $MEA(k)$ en tenant compte de la contrainte structurelle [resp. sans tenir compte de la contrainte structurelle], la quatrième colonne contient les énergies libres des structures $MEA(k)$ présentées dans la colonne précédente.

4.5.3 Algorithme

Algorithme 8 : l'algorithme de `RNAborMEA` qui recherche les structures $MEA(k)$ de la séquence w et de la structure s_0.

1: **Procédure** RNAborMEA(w, s_0, M)
2: Initialiser $M(i,j,k) = 0$ pour tous $1 \leq i \leq j \leq n$, $0 \leq k \leq n$.
3: Calculer $p_{i,j}$ pour tous $1 \leq i \leq j \leq n$ avec l'algorithme de McCaskill.
4: **pour** $i = 1$ **jusqu'à** n **faire**
5: $q_i = 1 - \sum_{j \neq i} p_{i,j}$
6: **fin pour**
7: **pour** $d = 0$ **jusqu'à** $n - 1$ **faire**
8: **pour** $i = 1$ **jusqu'à** $n - d$ **faire**
9: j = i + d
10: **pour** $k = 0$ **jusqu'à** K_{max} **faire**
11: **si** $j - i \leq \theta$ **alors**
12: **si** $k == 0$ **alors**
13: $M(i,j,k) = \sum_{r=i}^{j} \beta \cdot q_r$
14: **sinon**
15: interrompre
16: **fin si**
17: **sinon si** $j - i == \theta + 1$ **alors**
18: **si** $(i,j) \in s_0$ **alors**
19: $M(i,j,0) = 2 \cdot \alpha \cdot p_{i,j} + \sum_{r=i+1}^{j-1} \beta \cdot q_r$
20: $M(i,j,1) = \sum_{r=i}^{j} \beta \cdot q_r$
21: interrompre
22: **sinon**
23: $M(i,j,0) = \sum_{r=i}^{j} \beta \cdot q_r$
24: **si** $BP(i,j) == 1$ **alors**
25: $M(i,j,1) = 2 \cdot \alpha \cdot p_{i,j} + \sum_{r=i+1}^{j-1} \beta \cdot q_r$
26: **fin si**
27: interrompre
28: **fin si**
29: **sinon**
30: MAX_SCORE(w, s_0, i, j, M)
31: **fin si**
32: **fin pour**
33: **fin pour**
34: **fin pour**
35: **fin Procédure**

Algorithme 9 : la suite de l'algorithme de `RNAborMEA`

1: **Procédure** MAX_SCORE(w, s_0, i, j, M)
2: $max = 0$
3: ▷ Case 1 : j est non-apparié
4: $b_0 = D_{BP}(s_0[i, j-1], s_0[i, j])$
5: $val = M(i, j-1, k-b_0) + \beta \cdot q_j$
6: **si** $val > max$ **alors**
7: $max = val$
8: $index = (0, 0, 0)$
9: **fin si**
10: ▷ Case 2 : $(i, j) \in s$
11: **si** $BP(i, j) == 1$ **alors**
12: $b_1 = D_{BP}(s_0[i+1, j-1] \cup (i, j), s_0[i, j])$
13: $val = M(i+1, j-1, k-b_1) + 2 \cdot \alpha \cdot p_{i,j}$
14: **si** $val > max$ **alors**
15: $max = val$
16: $index = (i, k-b_1, 0)$
17: **fin si**
18: **fin si**
19: ▷ Case 3 : $(r, j) \in s, tel\ que\ i < r < j$
20: **pour** $r = i+1$ **jusqu'à** $j - \theta - 1$ **faire**
21: **si** BP(r,j) **alors**
22: $b_2 = D_{BP}(s_0[i, r-1] \cup s_0(r+1, j-1), s_0[i, j])$
23: **pour** $k_0 = 0$ **jusqu'à** $k - b_2$ **faire**
24: $k_1 = k - b_2 - k_0$
25: $val = M(i, r-1, k_0) + 2 \cdot \alpha \cdot p_{r,j} + M(r+1, j-1, k_1)$
26: **si** $val > max$ **alors**
27: $max = val$
28: $index = (r, k_0, k_1)$
29: **fin si**
30: **fin pour**
31: **fin si**
32: **fin pour**
33: $M(i, j, k) = max$
34: $M(j, i, k) = index$
35: **fin Procédure**

4.6 Résultats

Dans la section précédente, nous avons décrit le nouvel algorithme `RNAborMEA`, qui calcule pour une structure initiale arbitraire s_0, et pour tous les entiers k entre 0 et K_{max}, les structures $MEA(k)$ qui ont un score EA maximum parmi toutes les structures k-voisines de s_0.

Dans cette section, nous essayons d'abord de détecter les structures fonctionnelles d'une séquence du riboswitch TPP avec notre programme `RNAborMEA` (voir 4.6.1.1). Nous comparons ensuite les structures $MFE(k)$ avec les structures $MEA(k)$ en fonction de la distance de paires de bases (voir 4.6.1.2). Nous évaluons la prédiction des structures fonctionnelles de l'ensemble des séquences de la famille du riboswitch purine des six méthodes différentes (voir 4.6.1.3). Nous définissons à la fin une loi de pseudo-Boltzmann qui nous permet d'analyser les différents ensembles des structures k-voisines (voir 4.6.2).

4.6.1 Détection des structures fonctionnelles de Riboswitch

4.6.1.1 Riboswitch TPP

Nous exécutons d'abord notre programme `RNAborMEA` sur une molécule du roboswitch TPP, dont

- l'identification de la base de données EMBL est "AF269819/1811-1669".
- la séquence est "*CUACUAGGGGAGCCAAAAGGCUGAGAUGAAACCCUUAUAACC UGAUUUGGUUAAUACCAACGUAGGAAAGUAGUUAUUAACUAUUCGUCAUUG AGAUGUCUUGGUCUAACUACUUUCUUCGCUGGGAAGUAGUU*".
- la structure initiale est la structure MFE.

Nous obtenons à la sortie de `RNAborMEA`, l'ensemble de structures $MEA(k)$, ainsi que leur scores EA, et leur énergies libres. Nous les présentons en fonction de la distance de paires de bases dans la figure 4.6. Nous avons observé que la structure MEA(61) qui diffère 61 paires de bases de la structure initiale, a le score EA maximum parmi toutes les structures possibles de la séquence, y compris la structure initiale MEA(0). En même temps, nous avons remarqué que l'énergie libre de la structure MEA(61) est aussi faible que celle de la structure MFE. Nous montrons donc les structures MEA(0), MEA(61) dans la figure 4.7. Nous montrons également les structures fonctionnelles du riboswitch TPP de *B. subtilis* dans la figure 4.8. La structure du

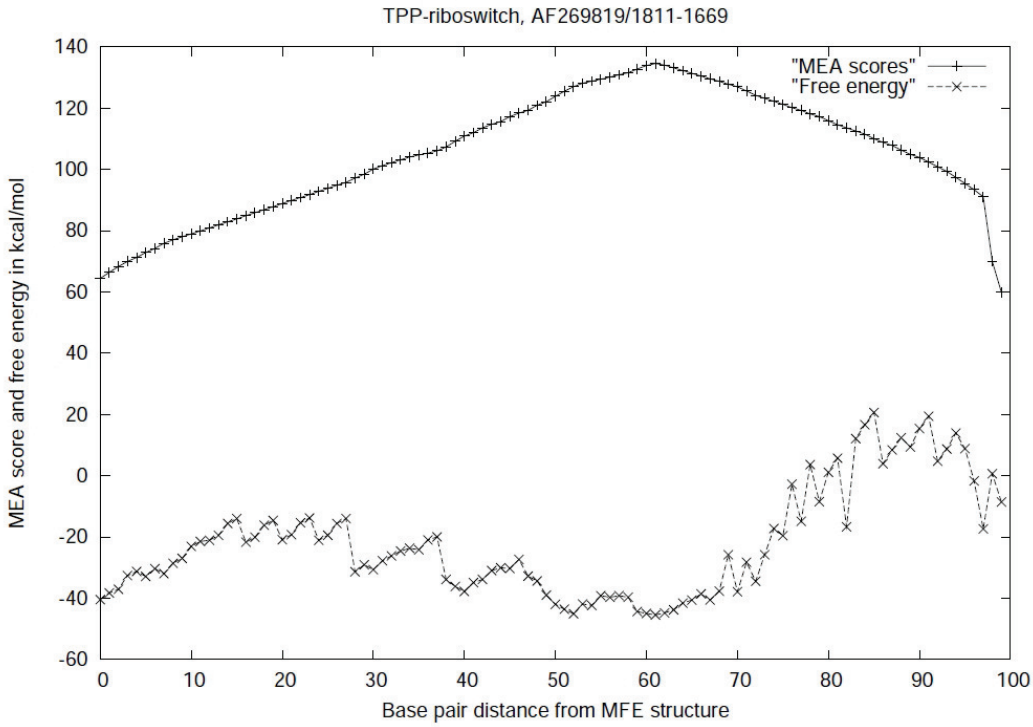


FIGURE 4.6 – Les score EA (la courbe en haut) et les énergies libres (la courbe en bas) des structures $MEA(k)$ du riboswitch TPP "AF269819/1811-1669".

"gène off" dans la figure 4.8 est la structure MFE prédite par `mfold` [109]. La structure du "gène on" dans la figure 4.8 est prédite par `mfold` en tenant compte de la contrainte structurelle qui interdit les appariements des paires de bases dans la tige terminatrice et qui permet de former la tige anti-terminatrice.

Dans cet exemple du riboswitch TPP, la structure MEA(61) détectée par `RNAborMEA` ressemblent bien à la structure du "gène on" du riboswitch TPP dans *B. subtilis*.

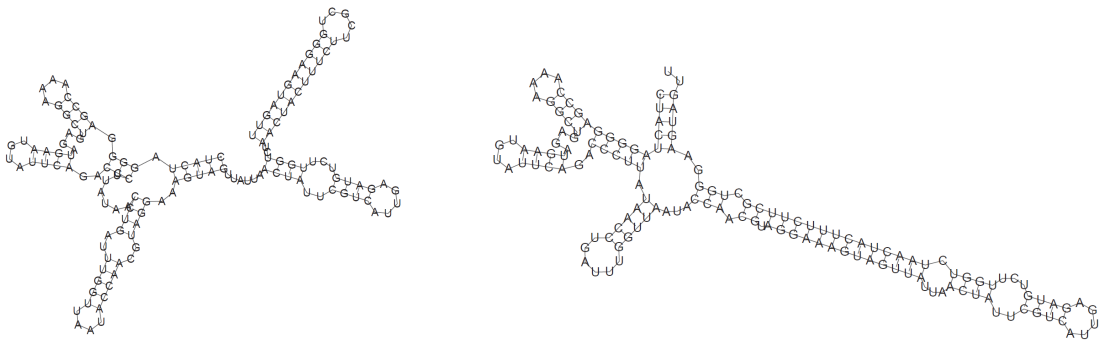


FIGURE 4.7 – Les structures MEA(0) (à gauche) et MEA(61) (à droite) du riboswitch TPP AF269819/1811-1669 détectées par `RNAborMEA`.

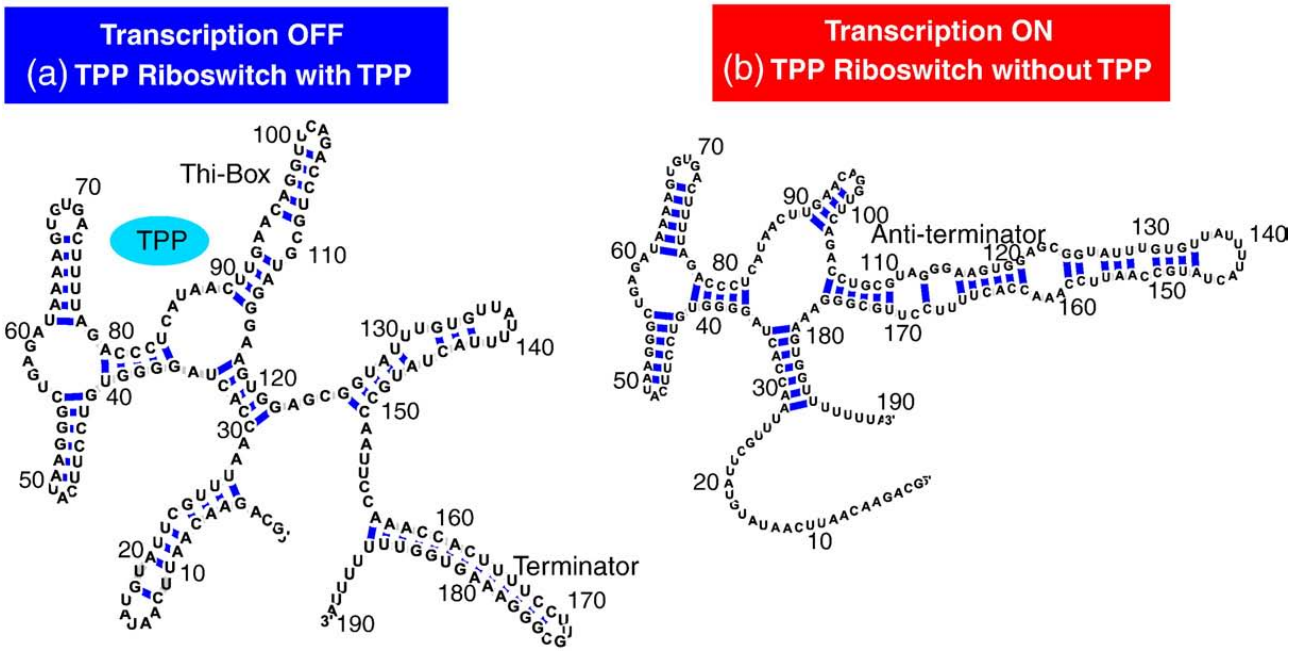


FIGURE 4.8 – Les structures du "gène on" et du "gène off" du riboswitch TPP dans *B. subtilis* [108].

4.6.1.2 Comparaison entre les structures $MFE(k)$ et les structures $MEA(k)$

Dans l'exemple précédent du riboswitch TPP, nous constatons que la structure du "gène off" est la structure MFE, et la structure du "gène on" est la structure MEA. Nous voulons donc savoir comment les structures $MEA(k)$ et les structures $MFE(k)$ sont différentes. Dans ce but, nous avons pris les 56 séquences de la famille "U7 small nuclear RNA" dont l'identifiant de Rfam est : $RF00066$. Pour chaque séquence w, nous exécutons les programmes `RNAbor` et `RNAborMEA` en prenant la structure MFE comme la structure initiale. Pour chaque distance de paires de bases k, nous aurons une structure MEF(k) $s_1(k)$ et une structure $MEA(k)$ $s_2(k)$. Nous calculons ensuite la distance de paires de bases d_k entre ces deux structures. Pour chaque distance de paires de bases, nous aurons donc 56 distances de paires de bases d_k pour les 56 séquences. La figure 4.9 nous montre pour chaque distance de paires de bases, la moyenne $\pm$ l'écart type des 56 distances de paires de bases.

La figure 4.9 montre que notre programme actuel `RNAborMEA` offre une manière différente des structures par rapport `RNAbor` : plus les structures sont éloignées de la structure MFE, plus la structure proposée par `RNAborMEA` est diffèrente de celle proposée par `RNAbor`.

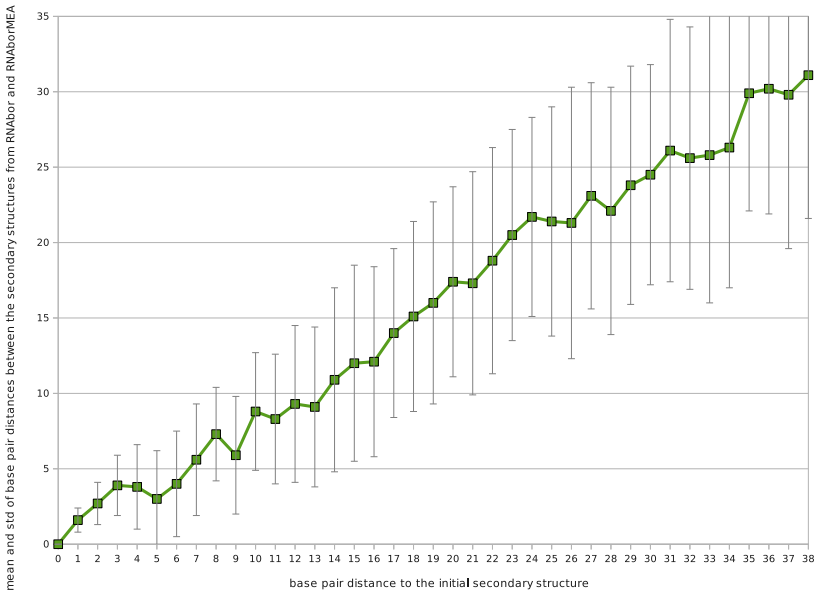


FIGURE 4.9 – Les moyennes et les écarts type des distances de paires de bases entre des structures $MFE(k)$ et des structures $MEA(k)$ en fonction de la distance de paires de bases de la structure intiale.

4.6.1.3 Riboswitch Purine

Les riboswitchs purine possèdent des structures d'ARN qui régulent la biosynthèse et le transport des nucléotides A ou G. Le riboswitch guanine XPT de *B. subtilis* est un riboswitch purine. Dans *Bacillus subtilis*, ce motif d'ARNm est situé sur au moins cinq unités transcriptionnelles différentes qui encodents les 17 gènes principalement impliqués dans le transport des purines et de la synthèse des nucléotides de purines.

Comme le mécanisme de contrôle de l'expression des gènes par le riboswitch guanine XPT de *B. subtilis* a été démontré expérimentalement [110], nous prenons ce riboswitch comme un benchmark :

- sa séquence est "*CACUCAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCG GGCACCGUAAAUGUCCGACUAUGGGUGAGCAAUGGAACCGCACGUGUACGG UUUUUUGUGAUAUCAGCAUUGCUUGCUCUUUAUUUGAGCGGGCAAUGCUUU UUUU*".
- ses deux structures du "gène on" et du "gène off" sont déterminées expérimentalement

[110] en utilisant la technique d'analyse "in-line probing" [111] sont affichées dans la figure 4.10.

– L'énergie libre du "gène on" (resp. du "gène off") est -16.46 kcal/mol. (resp. -22.6 kcal/mol). Les énergies libres sont calculées par le programme `RNAeval`.

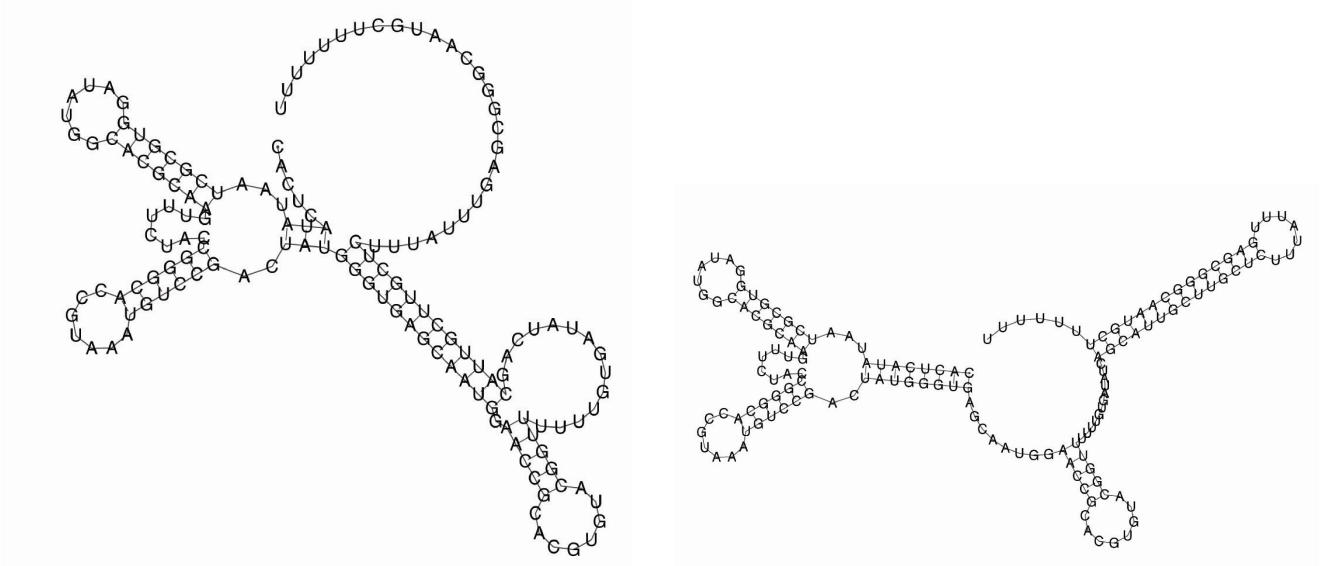


FIGURE 4.10 – Les structures secondaires du "gène on" (à gauche) et du "gène off" (à droite) du riboswitch guanine XPT de *B. subtilis*.

Nous appliquons 6 programmes différents, y compris notre programme `RNAborMEA`, sur toutes les séquences de la famille de riboswitch purine, pour prédire les deux structures fonctionnelles. Nous allons ensuite mesurer la qualité de ces structures prédites en comparant avec les structures déterminées expérimentalement du riboswitch guanine XPT de *B. subtilis* (voir la figure 4.10).

Pour chacune des séquences de la famille de riboswitch purine, nous itérons les étapes suivantes :

– obtenir la séquence de la région de l'aptamère à partir de la base de données Rfam [112].
– prédire la position qui indique la fin de la région de la plateforme d'expression.
– étendre la séquence jusqu'à la fin de la plateforme d'expression en utilisant la base de données EMBL.
– appliquer les 6 programmes `RNAbor` [71], `RNAborMEA` [113], `sampleRNAbor` [113], `RNAlocopt` [114], `RNAshapes` [70] et `UNAFold` [115] pour engendrer un ensemble des structures sous optimales dans lesquelles les deux structures fonctionnelles du "gène on" et du "gène off" sont susceptibles de se trouver.

– ces ensembles des structures sous-optimales sont :

`RNAborMEA` : les structures $MEA(k)$ $(0 \leq k \leq K_{max})$.

`RNAbor` **et** `sampleRNAbor` : les structures $MFE(k)$ $(0 \leq k \leq K_{max})$.

`RNAlocopt` : les structures localement optimales[5].

`RNAshapes` : les structures représentatives des classes de "shapes" [116] ayant leurs énergies dans un intervalle donné au dessus de la mininum énergie libre.

`UNAFold` : les structures sous-optimales générées avec l'algorithme de Zuker [132].

– pour chacun des 6 ensembles de structures sous-optimales, trouver une structure s_{on} la plus similaire à celle du gène "on" et une autre structure s_{off} la plus similaire à celle du gène "off" de riboswitch guanine XPT de B.subtilis.

Pour mesurer la similaritude structurelle entre deux structures d'ARN, nous utilisons le programme d'alignement structurel `NestedAlign` [117] qui donne un score d'alignement structurel. Plus le score est grand, plus les deux structures sont similaires.

Pour chacune des séquences de la famille de riboswitch purine, nous illustrons les scores de `NestedAlign` entre les structures s_{on} [resp. les structures s_{off}] des 6 programmes et la structure du "gène on" [resp. la structure du "gène off"] de riboswitch guanine XPT de *B.subtilis* (voir la figure 4.11 [resp. la figure 4.12]).

Dans les figures 4.11 et 4.12, l'axe des données X indique les identifiants des séquences, l'axe des données Y montre les scores de `NestedAlign` entre les structures prédites et les structures fonctionnelles du riboswitch guanine XPT de B.subtilis. Nous observons que parmi les 34 séquences testées, notre programme `RNAborMEA` a trouvé 21 structures la plus similaire à la structure du "gène on" et 22 structures la plus similaire à la structure du "gène off" du riboswitch guanine XPT de B.subtilis par rapport les 5 autres programmes. La plupart des structures prédites par `RNAborMEA` sont plus proches aux structures fonctionnelles des riboswitchs. D'autres parts, nous constatons que les qualités des prédictions de structure du "gène off" sont proches pour les six différents programmes, celles du "gène on" semblent signficativement supérieures pour notre programme.

5. une structure est localement optimale si son énergie libre ne peut pas diminuer en ajoutant ou supprimant une seule paire de base [114].

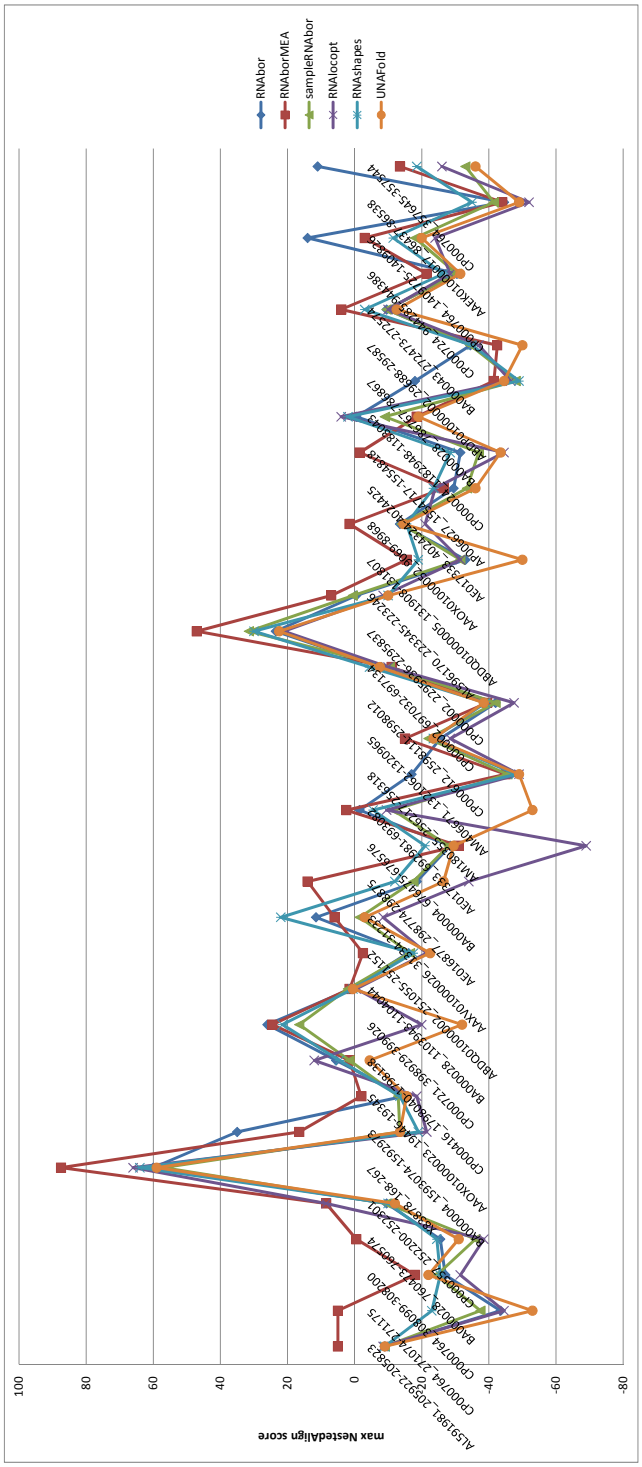


FIGURE 4.11 – La comparaison de la prédiction des structures du “gène on” de riboswitch purine des programmes `RNAborMEA` (rouge) `RNAbor` (bleu) `sampleRNAbor` (vert) `RNAlocopt` (violet) `RNAshapes` (turquoise) et `UNAFold` (orange).

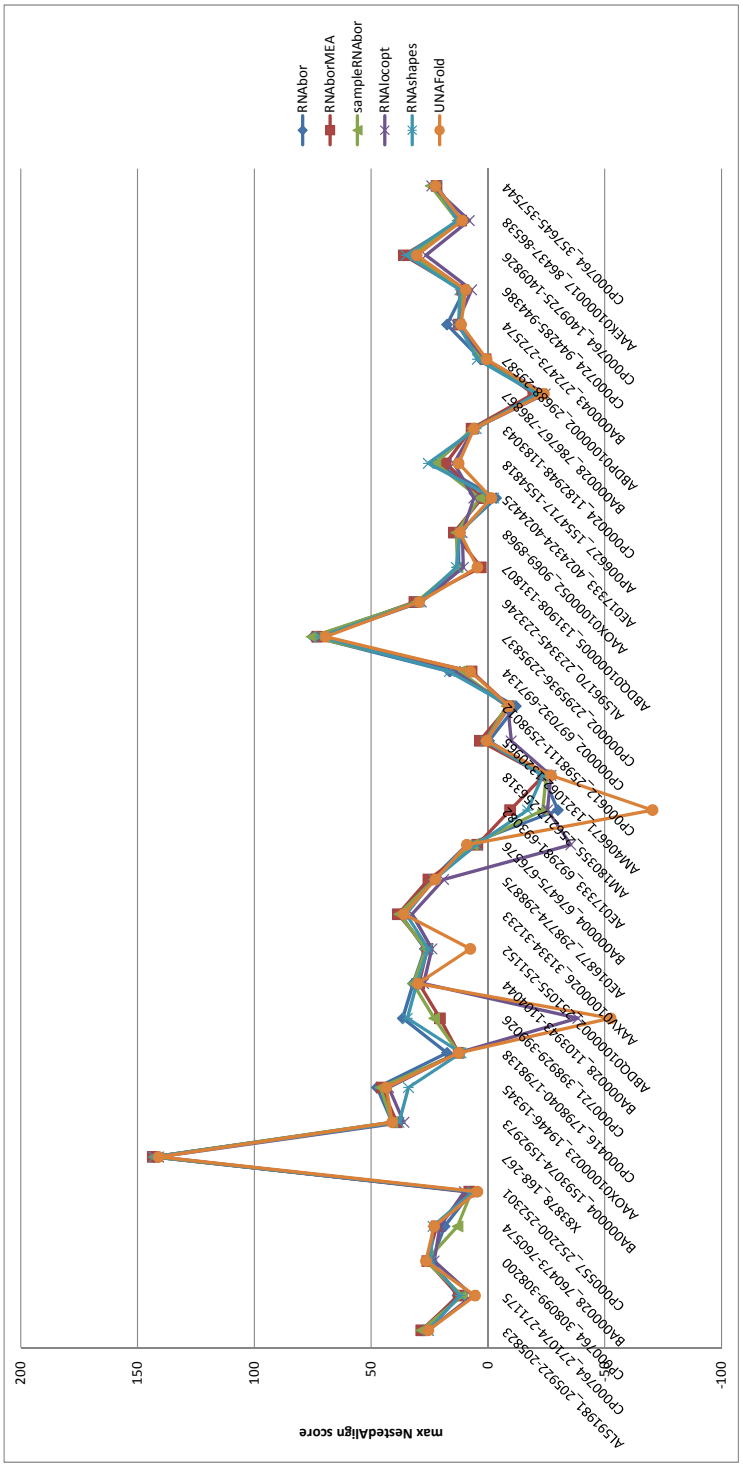


FIGURE 4.12 – La comparaison de la prédiction des structures du “gène off” de riboswitch purine des programmes `RNAborMEA` (rouge) `RNAbor` (bleu) `sampleRNAbor` (vert) `RNAlocopt` (violet) `RNAshapes` (turquoise) et `UNAFold` (orange).

4.6.2 Loi de pseudo-Boltzmann

Dans la section 4.3, nous avons montré une application du programme `RNAbor` [76], où les auteurs calculent les probabilités de Boltzmann des structures k-voisines d'un riboswitch SAM. Ils ont montré dans la figure 4.2 que la probabilité de Blotzmann à la distance de paires de bases 30 est maximum. Ils ont ensuite trouvé que la structure MFE(30) est proche de la structure fonctionnelle qu'on recherche.

L'analyse des probabilités de Boltzmann des structures k-voisines nous semble être une façon possible pour prédire les structures fonctionnelles de riboswitch. D'une manière similaire à l'algorithme de `RNAborMEA` nous avons défini une pseudo-fonction de partition de Boltzmann (pseudo-distribution de Boltzmann) en remplaçons le facteur de Boltzmann ($e^{E(s)/RT}$) par le pseudo-facteur de Boltzmann ($e^{EA(s)/RT}$) comme suit :

Soit S l'ensemble des structure de la séquence w de la taille n, s_0 une structre de w, S^k les structures k-voisines de s_0, R, T les paramètres prédéfinies.

Définition 4.7 *La **pseudo-fonction de partition de Boltzmann** $\tilde{Z}$ et la **pseudo-probabilité** des structures k-voisin $\tilde{p}_k$ suivent les formules suivantes :*

$$\begin{aligned} \tilde{Z} &= \sum_{s \in S} e^{EA(s)/RT} \\ \tilde{p}_k &= \frac{\sum_{s \in S^k} e^{EA(s)/RT}}{Z1} \end{aligned} \tag{4.7}$$

La pseudo-probabilité des structures k-voisines $\tilde{p}_k$ est donc la somme des pseudo-facteurs de Boltzmann des structures k-voisines divisée par la somme des pseudo-facteurs de Boltzmann de toutes les structures de w. Nous rappelons en même temps que la distribution uniforme N^k à la distance de paires de bases k est le nombre des structures k-voisines divisé par le nombre de toutes les structures de w.

Dans la figure 4.13, nous illustrons la pseudo-distribution de Boltzmann et la distribution uniforme des structures k-voisines en fonction de la distance de paires de bases pour la molécule *fdhA SECIS* dont :

- la séquence est *CGCCACCCUGCGAACCCAAUAAUAAAAUAUACAAGGGAGCAAG GUGGCG*.
- la structure initiale s_0 est "(((((((.(((...(((.....))).))).))))))).
- la valeur des paramètres $R \cdot T$ est égale à 49, qui est la taille de la séquence.

Nous avons observé que dans ce cas-là, la pseudo-distribution de Boltzmann est presque la même que la distribution uniforme des structures k-voisin. Nous choisissons la valeur de $R \cdot T$ soit égale à la taille de la séquence, parce que nous voulons normaliser les valeurs des pseudo-facteurs de Boltzmann entre 0 et 1.

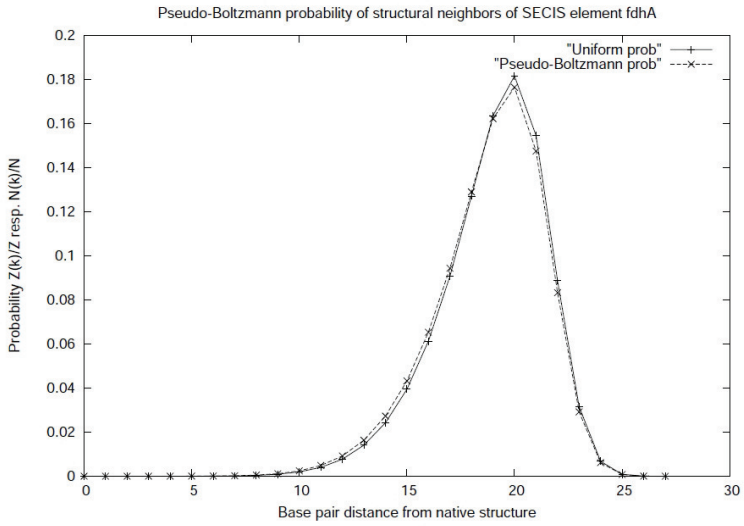


FIGURE 4.13 – La superposition de la pseudo-distribution de Boltzmann (la courbe $-\times-$) et la distribution uniforme (la courbe $-+-$) des structures k-voisines de la séquence *fdhA SECIS*.

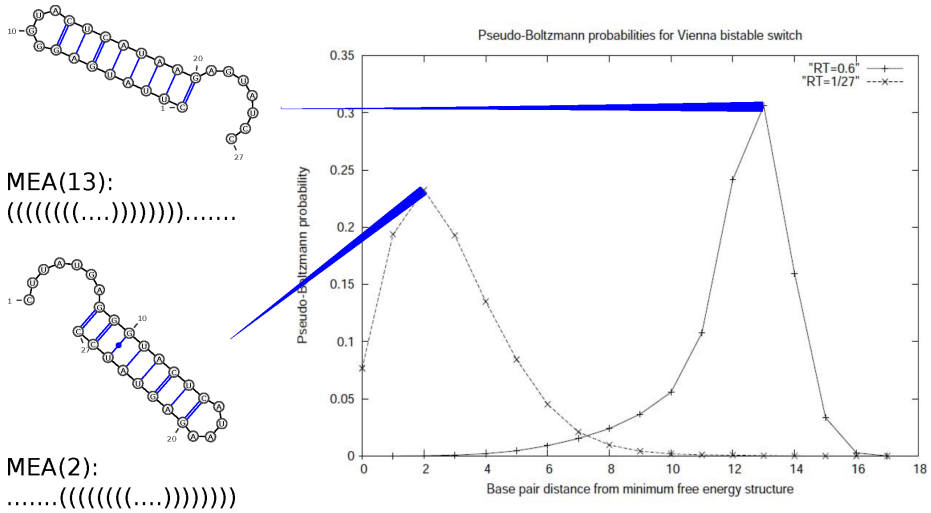


FIGURE 4.14 – Les pseudo-distributions de Boltzmann et les deux structures stables d'une molécule bistable générée par le groupe de Vienna Package.

Ensuite nous calculons la pseudo-distribution de Boltzmann sur une molécule bistable, dont la séquence est CUUAUGAGGG UACUCAUAAG AGUAUCC, la structure initiale s_0 est ".....(((((((....)))))))". Dans la figure 4.14, nous avons montré deux pseudo-distributions de Boltzmann dans lesquelles nous avons observé deux pics qui correspondent aux deux structures stables que nous recherchons. Quand la valeur de $R \cdot T$ est égale à 0.6, nous obtenons la courbe à gauche avec un pic à la distance de paires de bases 2, qui nous permet de trouver une des deux structures stable : la structure MEA(2) ".....(((((((....)))))))". Quand la valeur de $R \cdot T$ est égale à 27 qui est la taille de la séquence, nous obtenons la courbe à droite avec un pic à la distance de paires de bases 13 qui nous permet de trouver l'une autre des deux structures stable : la structure MEA(13) "(((((((....)))))))....."

4.7 Discussion

Dans ce chapitre, nous avons présenté un nouvel outil `RNAborMEA` qui recherche l'ensemble de structures $MEA(k)$ d'une séquence et d'une structure d'ARN donnée, nous calculons en même temps la pseudo-fonction de partition de Boltzmann pour ses structures k-voisines.

Nous avons d'abord recherché les structures $MEA(k)$ parmi des ensembles de structures k-voisines d'une séquence du riboswitch TPP. Nous avons observé que 1) la structure MEA(61) à la distance de paires de bases 61 de la structure MFE a le score EA maximum. 2) L'énergie libre de cette structure est aussi faible que la structure MFE. 3) La structure MEA(61) [la structure MFE] est bien similaire à la structure du "gène on" [la structure du "gène off"] du riboswitch TPP déterminée expérimentalement.

Nous avons ensuite appliqué notre programme `RNAborMEA` sur les 34 séquences de la famille du riboswitch purine. Nous avons alors observé que les structures $MEA(k)$, même si elles comportent des similaritudes aux structures fonctionnelles, ne sont pas celles attendues. Les motifs structuraux dans la région d'aptamère des riboswitchs purines sont bien prédits dans nos structures $MEA(k)$. Par contre, les motifs structuraux que nous avons trouvés dans la plateforme d'expression sont moins similaires aux celles que nous voulons prédire. Nous n'avons pas montré ces structures prédites dans ce chapitre. Une raison possible est qu'il existe des pseudo-noeuds dans les structures fonctionnelles des riboswitchs, les probabilités des paires de bases que nous utilisons pour calculer les scores EA ne tiennent pas compte de ces structures. Nous pensons que si nous pouvons tenir compte des structures avec les pseudo-noeuds dans notre programme `RNAborMEA`, la qualité de la prédiction des structures fonctionnelles des riboswitchs `RNAborMEA` peut être améliorée.

En prenant les mêmes 34 séquences de la famille de riboswitch purine comme les jeux d'essai, nous avons puis comparé les structures prédites par les six programmes différentes avec les structures attendues, nous avons observé que la plupart des structures prédites par notre programme `RNAborMEA` sont les structures le plus similaires structurellement aux structures fonctionnelles attendues.

Nous avons enfin défini la distribution de pseudo-Boltzmann sur les structures k-voisines, nous l'avons appliqué sur un exemple d'une séquence "bistalbe", les deux structures bistables sont obtenues en prennant les différentes valeurs des paramétres $R \cdot T$. Le point faible de cette méthode est qu'il faut choisir la valeur de $R \cdot T$, et en même temps, les valeurs possibles sont nombreuse. Il faut donc trouver la bonne valeur pour identifier la structure stable.

Une orientation possible dans la future pour l'améliorier notre programme est l'utilisation de la contrainte structurelle, il nous permet de diminuer le nombre de structures recherchées. Surtout dans notre cas des différentes familles de riboswitchs, nous connaissons déjà la structure

conservée dans la région de l'aptamère, nous pouvons la mettre dans la contrainte structurelle, ce qui nous permet de ne pas prédire des structures incorrectes dans la région de l'aptamère. De plus, David H. Mathews *et al.* [11] ont utilisé les contraintes des modifications chimiques qui sont été déterminées expérimentalement. Ils utilisent ces contraintes dans les algorithmes de programmation dynamique pour la prédiction de la structure secondaire d'ARN. Par conséquent, la précision de la prédiction de structure est considérablement améliorée.

Chapitre 5

Une nouvelle méthode de recherche d'alignements deux-à-deux sous-optimaux

5.1 Introduction

Le problème de l'alignement de séquences deux-à-deux est étudié depuis plus de quarante ans [15, 16, 118]. Dans les applications sur les séquences de protéines ou d'ADN, on recherche un alignement optimal, ayant un score mathématiquement maximal selon une fonction de score donnée, comme dans l'algorithme de Needleman-Wunsch [15] ou dans l'algorithme de Smith-Waterman [16].

Plus récemment, l'alignement de structures tertiaires de protéines est apparu, comme ce qui est produit par les logiciels DALI [119], CE [120], Topofit [121], etc. Sauder et ses collaborateurs [122] ont montré qu'il reste un écart important entre la précision de l'alignement de deux séquences et celle de l'alignement de deux structures tertiaires. En effet, pour un niveau d'identité de séquences de 10-15%, BLAST [17] aligne correctement 28% des paires de résidus, tandis que PSI-BLAST [18, 19] améliore la précision de l'alignement à 40%. Au même niveau d'identité de séquences, les programmes d'alignement structurel comme DALI [119] et CE [120], alignent correctement 75% des paires de résidus. La précision de l'alignement structurel est donc

bien meilleure que celle de l'alignement de séquences.

Je m'intéresse donc à la recherche d'alignement sous-optimaux de séquences pour améliorer la qualité d'alignement de séquences. Je suis motivé par les raisons suivantes : 1) on soupçonne que les alignements les plus pertinents biologiquement se trouvent plus fréquemment dans l'ensemble d'alignements sous-optimaux que dans l'alignement optimal. 2) Les alignements sous-optimaux de séquences peuvent produire les informations sur des paires de résidus alignés, qui peuvent être intégrées ensuite dans un modèle de régression logistique pour améliorer la qualité d'alignement[123].

J'ai donc développé et implémenté un nouvel algorithme, `SubOpt`, qui produit des alignements sous-optimaux à partir d'un alignement initial. J'ai fourni ensuite des résultats sur la performance d'alignement de `SubOpt` par rapport aux autres programmes existants.

5.2 Alignement de séquences

L'alignement de séquences est une manière de disposer les séquences d'ADN, d'ARN, ou de protéines pour identifier les régions similaires héritées d'un ancêtre commun. Il est utilisé dans beaucoup de problèmes en bioinformatique : identifier les sites fonctionnels, prédire la fonction d'une protéine, prédire la structure secondaire ou tertiaire d'une protéine ou d'un ARN, établir une phylogénie. On peut produire un alignement entre deux séquences (alignement deux-à-deux) ou entre plusieurs séquences (alignement multiple). Notre travail est uniquement consacré à l'alignement deux-à-deux.

Comparer deux séquences peut s'effectuer par un processus d'édition en considérant les 3 modifications élémentaires : (a) insertion : insertion d'un ou plusieurs résidus ; (b) délétion (ou suppression) : suppression d'un ou plusieurs résidus ; (c) substitution : remplacement d'un résidu par une autre. L'insertion et la suppression sont toutes deux représentées par un "gap". Chaque modification a un poids, dépendant de l'opération et des résidus en cause.

Un alignement $\mathbb{A}$ de deux séquences $a = a_1 a_2 \ldots a_n$ et $b = b_1 b_2 \ldots b_m$ est souvent représenté dans un tableau à deux lignes et à L colonnes :

$$\begin{array}{ccccc} a_1^* & a_2^* & a_3^* & \dots & a_L^* \\ b_1^* & b_2^* & b_3^* & \dots & b_L^* \end{array}$$

où a_i^* est un résidu a_x ou un gap $-$, b_i^* est un résidu b_y ou un gap $-$, $(a_i^*, b_i^*) \neq (-,-)$, $x \leq i$, $y \leq i$.

Un alignement $\mathbb{A}$ de deux séquences $a = a_1 a_2 \dots a_n$ et $b = b_1 b_2 \dots b_m$ est aussi un ensemble ordonné de paires des résidus ou des paires contenant un gap :

$$\mathbb{A} = \{(a_1^*, b_1^*), (a_2^*, b_2^*), (a_3^*, b_3^*), \dots, (a_L^*, b_L^*)\} \tag{5.1}$$

où

- (a_i^*, b_i^*) est une paire de résidus (a_x, b_y) ou une paire contenant un gap $(a_x, -)(-, b_y)$, $x \leq i$, $y \leq i$.
- Si on enlève tous les gaps dans l'ensemble des lettres $a_1 a_2 \dots a_L$ (resp. $b_1 b_2 \dots b_L$) et si on concatène toutes les lettres restants en respectant leur ordres, on obtient alors la séquence a (resp. b).

```
seq1                1 NLGPSTKDFGKISREFDNQ     19
                            ..:.||||:...::.
seq2                1 ----LERSFGKINMRLEDA     15
```

FIGURE 5.1 – Un exemple d'alignement de deux séquences.

La figure 5.1 illustre un alignement global entre deux séquences arbitraires de protéines dont les séquences sont : NLGPSTKDFGKISREFDNQ et LERSFGKINMRLEDA. La matrice de substitution utilisée est BLOSUM 62, la pénalité d'initiation d'un gap est égale à 10, la pénalité d'extension d'un gap est égale à 0.5. Nous verrons plus loin les notions de matrice de substitution et de pénalité de gap.

Dans cette figure,

- le symbole $-$ signifie un gap dans l'alignement.
- le symbole | indique que les deux résidus alignés sont identiques.
- les symboles **.** et **:** signifient une substitution d'un résidu par un autre résidu.

Dans cet alignement, il y a donc en total 11 substitutions et 4 insertions/délétions ou indel.

5.2.1 Score d'alignement

5.2.1.1 Formule de score

Pour rechercher un alignement optimal ou sous-optimal de deux séquences, nous avons d'abord besoin de définir une fonction qui calcule le score de similarité $S(\mathbb{A})$ d'un alignement $\mathbb{A}$ entre les deux séquences a et b. L'alignement optimal est l'alignement ayant le score maximal. Une des fonctions possibles est la suivante :

$$S(\mathbb{A}) = \sum_{(a_x^*,b_x^*)\in identités} f_{ident}(a_x^*) + \sum_{(a_y^*,b_y^*)\in substitutions} f_{sub}(a_y^*, b_y^*) + \sum_{(a_z^*,b_z^*),\ldots,(a_{z+n}^*,b_{z+n}^*)\in indels} W_g(n+1) \tag{5.2}$$

où

- $f_{ident}(a_x^*)$ est la fonction qui calcule le score d'une identité de résidu a_x^*.
- $f_{sub}(a_y^*, b_y^*)$ est la fonction qui calcule la pénalité d'une substitution de résidu a_y^* par un autre résidu b_y^*
- soit $n \geq 0$, $(a_z, b_z), \ldots, (a_{z+n}, b_{z+n})$ est un gap ou une région de gaps consécutifs, tel que : $a_z^* = \text{“}-\text{”}, \ldots, a_{z+n}^* = \text{“}-\text{”}, a_{z-1}^* \neq \text{“}-\text{”}, a_{z+n+1}^* \neq \text{“}-\text{”}$ ou $b_z^* = \text{“}-\text{”}, \ldots, b_{z+n}^* = \text{“}-\text{”}, b_{z-1}^* \neq' \text{‘}-\text{”}, b_{z+n+1}^* \neq \text{“}-\text{”}$.
- $W_g(n)$ est la fonction qui calcule la pénalité d'une région de n gaps consécutifs.

Le score de similarité $S(\mathbb{A})$ est donc une somme des scores pour chaque paire de résidus concordants alignés (identité) ou de résidus discordants alignés (substitution) plus les pénalités pour chaque ensemble de gaps continus (indels).

En pratique, on utilise généralement une matrice de substitution pour calculer les scores des identités et des substitutions des résidus, et une fonction de pénalité pour des régions de gap.

5.2.1.2 Matrice de substitution

Dans chaque cellule de la matrice de substitution, il y a une valeur de correspondance entre deux éléments. Dans le cas des protéines, la matrice est de taille 20 * 20, et dans le cas des

acides nucléiques, la matrice est de taille 4 * 4.

En utilisant la matrice de substitution, nous nous attendons à ce que dans les alignements prédits, les identités et les substitutions conservatives des résidus soient plus probablement apparues dans les alignements prédits que dans les alignements générés aléatoirement, et ainsi contribuent majoritairement à des grands scores. Inversement, nous nous attendons à ce que les substitutions non-conservatives des résidus soient moins probablement apparues dans les alignements prédits que dans les alignements générés aléatoirement, et ainsi contribuent majoritairement à des petits scores.

Ces matrices de substitution des acides aminés sont basées sur des substitutions observées entre les familles de protéines. Les matrices PAM (Point Accepted Mutation) ont été créées par Magaret Dayhoff et ses collaborateurs [126]. Elles ont été obtenues après l'alignement manuel d'environ 1300 séquences appartenant à 71 familles de protéines. Ce type de matrice est basé sur l'estimation de la probabilité que, suite à une mutation par substitution au cours de l'évolution, un acide aminé remplace un autre acide aminé sans que la fonction de la protéine ne soit altérée.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	12																				C
S	0	2																			S
T	-2	1	3																		T
P	-3	1	0	6																	P
A	-2	1	1	1	2																A
G	-3	1	0	-1	1	5															G
N	-4	1	0	-1	0	0	2														N
D	-5	0	0	-1	0	1	2	4													D
E	-5	0	0	-1	0	0	1	3	4												E
Q	-5	-1	-1	0	0	-1	1	2	2	4											Q
H	-3	-1	-1	0	-1	-2	2	1	1	3	6										H
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6									R
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5								K
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6							M
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5						I
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6					L
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4				V
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9			F
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10		Y
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

FIGURE 5.2 – La matrice de substitution PAM 250.

La figure 5.2 présente la matrice PAM250. Une valeur faible dans la matrice (par exemple : W

/ C = -8) signifie qu'il est peu probable d'observer la substitution d'un acide aminé par un autre (ici, d'un tryptophane par une cystéine) sans perte significative de la fonction de la protéine. Au contraire, une valeur forte (par exemple : Y / F = 7) signifie qu'il est probable d'observer la substitution d'un acide aminé par un autre (ici, d'une tyrosine par une phénylalanine).

Les matrices BLOSUM sont postérieures aux matrices PAM et ont été développées par Henikoff et Henikoff [127]. Les matrices BLOSUM sont construites à partir de 2000 "blocs" provenant de plus de 500 familles de protéines. Les "blocs" sont des régions conservées de familles de protéines ne contenant pas d'insertion ni de délétion.

5.2.1.3 Pénalités de gap

Étant donné un gap de longeur n, la fonction de pénalité W_g peut être calculée de plusieurs façons différentes :

- **pénalité linéaire** :

$$W_g = d * n$$

  où d est la pénalité élémentaire.

- **pénalité affine** :

$$W_g = \alpha + (n-1) * \beta$$

  où α désigne la pénalité d'ouverture d'un gap et β est la pénalité d'extension d'un gap de longueur 1.

- **pénalité logarithmique** :

$$W_g = \alpha + \ln(n-1) * \beta$$

- **pénalité log-affine** :

$$W_g = \alpha + \ln(n-1) * \beta + \gamma$$

Nous savons qu'une séquence biologique est beaucoup plus susceptible d'avoir un grand gap, en raison d'une seule insertion ou suppression, que d'avoir plusieurs petits gaps. Or la pénalité linéaire d'une région de gaps de longueur 10 est la même que la pénalité linéaire de 10 gaps

de longueur 1. Donc la pénalité linéaire n'est pas appropriée pour l'alignement de séquences biologiques.

Dans [128], Reed A. Cartwright et ses collaborateurs ont calculé des alignements globaux sur un ensemble de paires de séquences en utilisant les pénalités affine, logarithmique et log-affine. Les précisions d'alignements ont été calculées en les comparant avec des alignements de référence construits par des experts. Les différentes fonctions de pénalité de gap ont été comparées en utilisant la moyenne des précisions d'alignements de l'ensemble de paires d'alignements testés. Ils ont montré que la pénalité log-affine de gap a la meilleure précision, suivie de près par la pénalité affine de gap, et que la pénalité logarithmique de gap a des résultats médiocres.

En pratique, la pénalité affine de gap est la plus utilisée dans les algorithmes existants d'alignement de protéines. Les raisons sont les suivantes : 1) la précision d'alignement est presque aussi bonne que pour la pénalité log-affine. 2) la pénalité d'une région de gaps consécutifs peut être décomposée en un ensemble des pénalités d'un gap par des algorithmes de programmation dynamique avec la complécité $O(n^2)$, et pour la pénalité log-affine, la complexité est $O(n^3)$.

5.2.2 Alignement global et local

Les alignements sont toujours effectués entre deux séquences complètes sous des conditions différentes.

- L'alignement peut être global, c'est-à-dire qu'on essaie d'aligner tous les résidus des deux séquences, d'identifier les régions conservées et les différences entre ces deux séquences. (voir la figure 5.3). L'algorithme le plus connu pour l'alignement global est l'algorithme de Needleman-Wunsch [15]. Le score de similaritude est calculé sur l'alignement complet.

```
seq1                1 NLGPSTKDFGKISESREFDNQ     21
                          ..:.||||:...|..
seq2                1 ----LERSFGKINMRLEDA--     15
```

FIGURE 5.3 – un alignement global de score maximal entre deux séquences arbitraires dont les séquences sont : NLGPSTKDFGKISREFDNQ et LERSFGKINMRLEDA. La matrice utilisée est BLOSUM 62, les pénalité d'ouverture et d'extension d'un gap sont 10 et 0.5. L'alignement est généré par le serveur web "EMBOSS Needle du site EBI

– L'alignement peut être local, c'est-à-dire qu'on essaie de voir si une région dans une séquence s'aligne bien avec une région dans l'autre séquence. (voir la figure 5.4). Il est utilisé quand les séquences sont dissemblables et susceptibles de contenir des régions ou des motifs similaires. L'algorithme le plus connu pour l'alignement local des séquences est l'algorithme de Smith-Waterman [16].

```
seq1                7 KDFGKIS     13
                      :.||||:
seq2                3 RSFGKIN      9
```

FIGURE 5.4 – un alignement local de score maximal entre deux séquences arbitaires : NLGPSTKDFGKISREFDNQ et LERSFGKINMRLEDA, , la matrice utilisée est BLOSUM 62, les pénalité d'ouverture et d'extension d'un gap sont 10 et 0.5. L'alignement est généré par le serveur web "EMBOSS Needle"

La partie précédente nous permet de calculer le score de n'importe quel alignement deux-à-deux, ce qui nous reste est de trouver l'alignement optimal avec le score maximal. Nous pouvons utiliser, par exemple, les algorithmes de type programmation dynamique, comme l'algorithme de Needleman-Wunsch pour l'alignement global [15], celui Smith-Waterman [16] pour l'alignement local et celui de Gotoh [118] pour l'alignement global ou local en utilisant la pénalité affine de gap. Plus de détails sur ce type d'algorithmes vont être discutés dans la sous section 5.3.2 à la page 96.

5.2.3 Algorithme d'alignements sous-optimaux

Dans cette-sous section, nous présentons quelques algorithmes existants pour la génération des alignements sous-optimaux. Ces alignements peuvent être générés dans un intervalle de score qui est proche du score optimal, comme l'algorithme de Waterman et Eggert [129] et l'algorithme de Zuker [132]. Ces alignements peuvent être engendrés par une matrice des probabilités $p(a_i, b_j)$ des paires de résidus de deux séquences, comme le programme `probA` [133].

Waterman est la première personne qui a considéré le problème de l'alignement sous-optimal. Dans [129], il a décrit comment modifier la formule de récurence standard, afin de générer tous les alignements dont le score dépasse un seuil proche du score maximal et défini par l'utilisateur.

Le problème avec cette méthode en pratique, est qu'il y a un nombre énorme d'alignements sous-optimaux qui ne s'écartent que très légèrement de l'alignement optimal et dont le score est presque aussi grand que celui de l'alignement optimal. Dans [130], Waterman et Eggert ont décrit comment générer un premier alignement sous-optimal (local) qui est produit en ne permettant l'alignement d'aucune des paires de résidus trouvés dans l'alignement optimal. Un deuxième alignement sous-optimal (local) est trouvé en ne permettant l'alignement d'aucune des paires de résidus de l'alignement optimal et le premier alignement sous-optimal, *etc.*

Zuker a ensuite décrit un algorithme [132] qui, comme l'algorithme de Waterman et Eggert, prédit un ensemble d'alignements sous-optimaux qui ont les scores d'alignement dans un intervalle proche du score optimal et qui contiennent certains paires de résidus a_i, b_j qui se trouvent dans l'alignement optimal. Un programme correspondant à cet algorithme est actuellement disponible sur le site web de `noptalign` : `http://fasta.bioch.virginia.edu/noptalign/`.

Différent des deux algorithmes précédents, le programme `probA` [135, 136] calcule d'abord la fonction de partition $Z(T)$ de tous les alignements globaux des deux séquences a et b comme suit :

Soit $\mathcal{A}$ l'ensemble des alignements des séquences a et b.

$$Z(T) = \sum_{\mathbb{A} \in \{\mathcal{A}\}} e^{\beta * S(\mathbb{A})}$$

où T est un paramètre inspiré de la distribution de Boltzmann, β est une constante, $S(\mathbb{A})$ est le score de similarité de l'alignement $\mathbb{A}$ qui dépend de la valeur de T.

La fonction de partition $Z(T)$ est ensuite utilisée pour calculer la probabilité de chaque paire possible de résidus et pour générer des nouveaux alignements optimaux et sous-optimaux correctement pondérés, de sorte que la probabilité d'un alignement $\mathbb{A}$ est :

$$Prob(\mathbb{A}) = \frac{1}{Z(T)} * e^{\beta * S(\mathbb{A})}$$

Le programme `probA` est disponible sur le site `http://www.tbi.univie.ac.at/~ulim/probA/`.

5.3 Méthode de `SubOpt`

La méthode `SubOpt` développée dans le cadre de cette thèse permet de générer un ensemble d'alignements sous-optimaux de deux séquences de protéines à partir d'un alignement initial. Nous définissons d'abord deux notions : la distance entre deux alignements, et le k-alignement.

5.3.1 Distance entre deux alignements

Soient $a = (a_1, \ldots, a_n)$, $b = (b_1, \ldots, b_m)$ deux séquences de lettres de tailles repectives n et m. Soit $\mathbb{A}_0$ un alignement arbitraire qui peut être un alignement produit par BLAST [134], l'algorithme de Needleman-Wunsch, l'algorithme de Smith-Waterman, n'importe quel alignement partiel produit manuellement, ou même un alignement vide [1]. Soit $\mathbb{A}_1$ un autre alignement des séquences a et b. Nous définissons quelques notions comme suit :

Définition 5.1 $\boldsymbol{P_L(\mathbb{A}_1)}$ *est l'ensemble de paires de lettres alignées dans l'alignement* $\mathbb{A}_1$.

Définition 5.2 $\boldsymbol{D_A(\mathbb{A}_0, \mathbb{A}_1)}$ *est la distance entre les alignements* $\mathbb{A}_0$ *et* $\mathbb{A}_1$*, qui est le nombre de paires d'acides aminés alignés dans* $P_L(\mathbb{A}_0)$ *et pas dans* $P_L(\mathbb{A}_1)$ *plus le nombre de paires d'acides aminés alignés dans* $P_L(\mathbb{A}_1)$ *et pas dans* $P_L(\mathbb{A}_0)$ *:*

$$D_A(\mathbb{A}_0, \mathbb{A}_1) = Card(P_L(\mathbb{A}_0) \backslash P_L(\mathbb{A}_1)) + Card(P_L(\mathbb{A}_1) \backslash P_L(\mathbb{A}_0))$$

Définition 5.3 *Un* k***-alignement*** *de* $\mathbb{A}_0$ *est un alignement ayant le score maximal parmi l'ensemble des alignements à distance d'alignement* k *de* $\mathbb{A}_0$.

Voici un exemple pour calculer la distance entre deux alignements.

Soient a, b les deux séquences : a = NLGPSTKDFGKISREFDNQ, b = LERSFGKINMRLEDA,

Soit $\mathbb{A}_0$ un alignement des séquences a et b :

1. Un alignement vide est un alignement ne contenant aucune paire de résidus : tous les résidus sont alignés avec un gap

```
NLGPSTKDFGKISREFDNQ
L----ERSFGKINMRLEDA
```

Soit $\mathbb{A}_1$ l'autre alignement des séquences a et b :

```
NLGPSTKDFGKISREFDNQ
----LERSFGKINMRLEDA
```

Alors,

$P_L(\mathbb{A}_0)$ = {(N,L), (T,E), (K,R), (D,S), (F,F), (GG,), (K,K), (I,I), (S,N), (R,M), (E,R), (F,L), (D,E), (N,D), (Q,A)}.

$P_L(\mathbb{A}_1)$ = {(S,L), (T,E), (K,R), (D,S), (F,F), (GG,), (K,K), (I,I), (S,N), (R,M), (E,R), (F,L), (D,E), (N,D), (Q,A)}.

$D_A(\mathbb{A}_0, \mathbb{A}_1) = Card(\{(N, L)\}) + Card(\{(S, L)\}) = 1 + 1 = 2.$

5.3.2 Méthode de `SubOpt`

5.3.2.1 Entrées et Sorties

Les éléments nécessaires pour exécuter `SubOpt` sont :

- Deux séquences : $a = (a_1, a_2, \ldots, a_n)$, $b = (b_1, b_2, \ldots, b_m)$.
- une matrice de substitution : qui peut être la matrice PAM, BLOSUM, ...
- une fonction de pénalité de gaps : qui peut être la pénalité linéaire, la pénalité affine, ...
- un alignement initial $\mathbb{A}_0$: qui peut être un alignement global, local, ou vide.
- un entier K_{max} : qui définit la distance maximale entre 2 alignements. Il peut être par exemple la taille d'une des deux séquences.

À partir de ces données, `SubOpt` calcule :

- $(K_{max} + 1)$ k-alignements : $\mathbb{A}_k$, tels que $0 \leq k \leq K_{max}$.

Pour chaque distance d'alginement k, il peut y avoir plusieurs k-alignements qui ont le mème score maximal. En partique, nous gardons un seul k-alignement qui est le premier trouvé en "back track" dans la matrice M, de façon similaire à l'algorithme de Needleman-Wunsch [15].

5.3.2.2 Matrice de score d'alignement

Nous avons implémenté notre programme `SubOpt` par programmation dynamique avec la fonction de pénalité linéaire et aussi affine de gaps. Ici, nous allons juste montrer comment calculer les scores optimaux pour toutes les sous-séquences et pour toutes les distances d'alignement avec la pénalité linéaire de gaps. Pour la pénalité affine de gaps, c'est similaire à l'algorithme de Gotoh [118] : au lieu d'utiliser une matrice de score, nous utilisons trois matrices pour différencier l'ouverture et l'extension de gap.

Nous définissons une matrice de score M de taille $(n+1)\times(m+1)\times(K_{max}+1)$, où $M(i,j,k)$ est le score maximal d'alignement parmi tous les alignements des sous-séquences $a(1,i)$, $b(1,j)$ à distance d'alignement k de $\mathbb{A}_0$, tel que $a(1,i)$ (resp. $b(1,j)$) est la partie de séquence entre la position 1 et i (resp. 1, j).

Nous initialisons les valeurs de la matrice M, où soit $i=0$ soit $j=0$, de la façon suivante :

$$M(i,j,k) = \begin{cases} 0 & \text{si } i=j=k=0 \\ -\infty & \text{si } i=j=0 \text{ et } k>0 \\ g\cdot j & \text{si } i=0 \text{ et } k=0 \\ -\infty & \text{si } i=0 \text{ et } k>0 \\ g\cdot i & \text{si } j=0 \text{ et } k=0 \\ -\infty & \text{si } j=0 \text{ et } k>0. \end{cases} \tag{5.3}$$

où g est la pénalité d'un gap dans la fonction de pénalité linéaire.

Les autres valeurs dans la matrice M vont être calculées comme suit.

$$M(i,j,k) = \max \begin{cases} M(i-1,j-1,k) + \sigma(a_i,b_j) & \text{si } (i,j) \in \mathbb{A}_0 \\ M(i-1,j-1,k-2) + \sigma(a_i,b_j) & \text{si } (i,-),(r,j) \in \mathbb{A}_0, \text{ for some } 1 \leq r < i \\ M(i-1,j-1,k-2) + \sigma(a_i,b_j) & \text{si } (i,r),(-,j) \in \mathbb{A}_0, \text{ for some } 1 \leq r < j \\ M(i-1,j-1,k-1) + \sigma(a_i,b_j) & \text{si } (i,-),(-,j) \in \mathbb{A}_0 \\ M(i,j-1,k) + g & \text{si } (-,j) \in \mathbb{A}_0 \\ M(i,j-1,k-1) + g & \text{si } (r,j) \in \mathbb{A}_0, \text{ for some } 1 \leq r \leq i \\ M(i-1,j,k) + g & \text{si } (i,-) \in \mathbb{A}_0 \\ M(i-1,j,k-1) + g & \text{si } (i,r) \in \mathbb{A}_0, \text{ for some } 1 \leq r \leq j \end{cases} \tag{5.4}$$

où $\sigma(a_i, b_j)$ est le score de la paire de résidus (a_i, b_j) dans la matrice de substitution.

Voici l'explication de cette formule de récurrence.

Pour les lettres a_i et b_j, il existe trois possibilités comme suit :

1. Soit l'acide aminé à la position i de la séquence a est aligné avec celui à la position j de la séquence b :

FIGURE 5.5 – Un exemple d'alignement où la lettre a_i aligne avec la lettre b_j.

   – Si a_i est aligné avec b_j dans l'alignement $\mathbb{A}_0$ (alignement initial), alors

     $M(i,j,k) = M(i-1,j-1,k) + \sigma(a_i,b_j)$,

   – Si a_i est aligné avec un gap, b_j est aligné avec a_r dans l'alignement $\mathbb{A}_0$, où $1 \leq i$, alors

     $M(i,j,k) = M(i-1,j-1,k-2) + \sigma(a_i,b_j)$

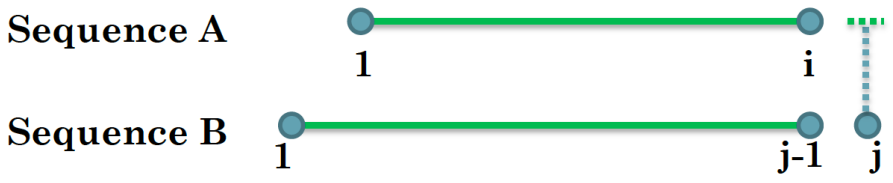
   – Si a_i est aligné avec b_r, b_j est aligné avec un gap dans l'alignement $\mathbb{A}_0$, alors

     $M(i,j,k) = M(i-1,j-1,k-2) + \sigma(a_i,b_j)$

   – Si a_i et b_j sont tous alignés avec un gap dans l'alignement $\mathbb{A}_0$, alors

$M(i,j,k) = M(i-1,j-1,k-1) + \sigma(a_i, b_j)$

2. Soit l'acide aminé à la position j de la séquence b est aligné avec un gap de la séquence a :



   – Si b_j est aligné avec un gap dans l'alignement $\mathbb{A}_0$, alors

     $M(i,j,k) = M(i,j-1,k) + g,$

   – Si b_j est aligné avec a_r dans l'alignement $\mathbb{A}_0$, où $1 \leq r \leq i$, alors

     $M(i,j,k) = M(i,j-1,k-1) + g$

3. Soit l'acide aminé à la position i de la séquence a aligne avec un gap de la séquence b :

   Dans l'alignement $\mathbb{A}_0$,

   – Si a_i est aligné avec un gap dans l'alignement $\mathbb{A}_0$, alors

     $M(i,j,k) = M(i-1,j,k) + g,$

   – Si a_i est aligné avec b_r dans l'alignement $\mathbb{A}_0$, où $1 \leq r \leq< i$:

     $M(i,j,k) = M(i-1,j,k-1) + g$

5.3.2.3 k-alignement

Aprés avoir effectué le calcul de tous les scores dans la matrice M, nous aurons, pour $0 \leq k \leq K_{max}$, les scores des k-alignements dans les cases $M(n, m, k)$. Nous retrouvons les k-alignements en "back-track" avec la matrice M.

5.4 Résultats

Dans la partie des résultats, nous avons d'abord comparé la qualité de nos alignements sous-optimaux à celles des alignements optimaux produit par l'algorithme de Needleman-Wunsch. Ensuite, nous avons comparé les alignements sous-optimaux produits par les trois méthodes différentes.

5.4.1 *BAliBASE*

Pour évaluer notre programme d'alignement sous-optimal des séquences de protéines `SubOpt`, nous avons besoin d'un grand nombre d'alignements de référence. *BAliBASE* est une base de données d'alignements multiples de référence raffinés manuellement dont les alignements sont classés notamment par la longueur et le taux d'identité des séquences. Les régions qui sont jugées alignées [2] de façon particulièrement fiable sont indiquées sur les alignements multiples.

Selon les études dans [122], les algorithmes d'alignement de séquences ne sont pas efficaces quand les identités de séquences sont faibles. Donc, nous avons choisi 10 alignements multiples à partir de *BAliBASE* dont les identités de séquences sont de moins de 25%, les identifiants sont : 1aboA (voir la figure 5.6), 1csp, 1dox, 1fkj, 1fmb, 1krn, 1plc, 2fxb, 2mhr,9rnt. À partir de ces 10 alignements multiples, 92 alignements deux à deux sont obtenus. Ils vont être utilisés dans nos tests.

5.4.2 Comparaison avec l'alignement global

La table 5.1 donne le nombre d'alignements trouvé par `SubOpt` en fonction de la distance d'alignement sur les 92 alignements de référence, en utilisant la matrice de substitution BLOSUM45. Dans les tests des tableaux 5.1 et 5.3, la pénalité d'initiation d'un gap est égale à -14, la pénalité d'extension d'un gap est égal à -2, l'alignement initial est l'alignement global produit par l'algorithme de Needleman-Wunsch. Pour chaque distance k, il peut y avoir un

2. Ces régions sont appelées "core blocks" en anglais.

1aboA_ref1 - reference 1

```
Name                    SH3

Number of sequences                 5
Alignment Length                    88
Longest Sequence                    74
Shortest Sequence                   48
Average Percent Identity            18
Maximum Percent Identity            26
Minimum Percent Identity            14

Sequence Name     SWISSPROT Accession
1aboA                 P00520
1ycsB                 P04637
1pht                  P27986
1ihvA                 P00383
1bb9                  O00499

Family    1aboA 1ycsB 1pht
Family    1ihvA
Family    1bb9


1aboA        1  .NLFVALYDfvasgdntlsitkGEKLRVLgynhn...............gE
1ycsB        1  kGVIYALWDyepqnddelpmkeGDCMTIIhrede............deiE
1pht         1  gYQYRALYDykkereedidlhlGDILTVNkgslvalgfsdgqearpeeiG
1ihvA        1  .NFRVYYRDsrd......pvwkGPAKLLWkg..................eG
1bb9         1  mFKVQAQHDytatdtdelqlkaGDVVLVIpfqn..........peeqdeG

1aboA       36  WCEAQt..........kngqGWVPSNYITPVN......
1ycsB       39  WWWARl..........ndkeGYVPRNLLGLYP......
1pht        51  WLNGYnet........tgerGDFPGTYVEYIGrkkisp
1ihvA       27  AVVIQd..........nsdiKVVPRRKAKIIRd.....
1bb9        41  WLMGVkesdwnqhkelekcrGVFPENFTERVQ......
```

Key

```
alpha helix         RED
beta strand         GREEN
core blocks         UNDERSCORE
```

FIGURE 5.6 – L'alignement multiple $1aboA_ref1$ extrait à partir de *BAliBASE*.

certain nombre de k-alignements qui ont le même score maximal. Nous considérons un seul k-alignement qui est le premier trouvé en "back track" par `SubOpt`.

La table 5.1 montre que sur les 92 alignements de référence, 15 alignements peuvent être trouvés par l'algorithme Needleman-Wunsch et aussi par l'algorithme `SubOpt`, à distance d'alignement 0, mais `SubOpt` peut trouver 18 alignements de plus où la distance d'alignement varie de 2 à 40. Si aucun alignement de référence n'est trouvé pour la distance d'alignement k, alors la valeur k n'est pas affichée dans la colonne de gauche de la table 5.1.

La table 5.2 montre le nombre d'alignements trouvés parmi l'ensemble des k-alignements en fonction des paramètres d'initialisation d'un gap α et d'extension d'un gap β. Comme précédemment, nous considérons un seul k-alignement pour chaque distance d'alignement.

Nous avons observé que quand $\alpha = -14$ et $\beta = -6$, `SubOpt` a prédit le plus grand nombre d'alignements parmi les 92 alignements de référence. De plus, pour la plupart des valeurs possibles de α et β ($-8 \leq \alpha \leq -16, -2 \leq \beta \leq -6$), les nombres d'alignements de référence trouvés par `SubOpt` vont de 27 jusqu'à 35.

k	nombre d'alignements
0	15
2	3
4	3
6	2
8	2
9	2
12	1
19	1
23	1
30	1
36	1
40	1
TOTAL	15+18=33

TABLE 5.1 – Nombre d'alignements de *BAliBASE* trouvés parmi l'ensemble des k-alignements de `SubOpt` en fonction de la distance d'alignement.

$\beta \backslash \alpha$	-2	-4	-6	-8	-10	-12	-14	-16
-1	3	9	23	31	29	30	33	31
-2	4	11	27	31	31	32	33	30
-3	3	14	28	31	33	31	31	31
-4	3	13	24	30	32	33	31	33
-5	3	11	24	27	31	32	32	33
-6	3	11	22	27	30	33	**35**	33

TABLE 5.2 – Nombre d'alignements de *BAliBASE* trouvés parmi l'ensemble des k-alignements de `SubOpt` en fonction des paramètres d'initialisation d'un gap α et d'extension d'un gap β.

Soit $\mathbb{A}_1$, $\mathbb{A}_2$ deux alignements prédits par deux algorithmes différents, soit $\mathbb{A}_0$ l'alignement de référence, on dit l'alignement $\mathbb{A}_1$ surpasse $\mathbb{A}_2$ si $D_A\{\mathbb{A}_1, \mathbb{A}_0\} < D_A\{\mathbb{A}_2, \mathbb{A}_0\}$.

La table 5.3 montre le pourcentage des alignements de référence dans lesquels `SubOpt` surpasse l'algorithme de Needleman-Wunsch, en fonction de la distance d'alignement k. Nous ne

montrons pas la distance $k = 0$, car dans ce cas-là, les résultats sont les mêmes pour ces deux algorithmes. Pour mesurer la similarité entre les deux alignements, nous utilisons justement la distance d'alignement : plus la distance est petite, plus les deux alignements sont similaires.

Nous avons observé que la majorité des k-alignements prédits par l'algorithme de `SubOpt` sont plus similaires aux alignements de référence que ceux prédits par Needleman-Wunsch. Par exemple, quand la distance d'alignement est égale à 4, 65.1% des 4-alignements prédits par `SubOpt` sont plus similaires aux alignements de référence que les alignements globaux prédits par l'algorithme de Needleman-Wunsch.

k	`SubOpt`
1	50.5%
2	58.9%
3	52.1%
4	65.1%
5	53.2%
6	67.3%
7	55.8%
8	64.7%
9	55.2%
10	65.0%
11	53.7%
12	64.6%

TABLE 5.3 – Pourcentage des alignements de référence dans lesquels notre programme `SubOpt` surpasse l'algorithme de Needleman-Wunsch.

5.4.3 Comparaison des 3 méthodes d'alignement sous-optimal

Dans cette sous-section, nous allons comparer les alignements sous-optimaux prédits par trois méthodes différentes.

1. l'échantillonnage d'alignements sous-optimaux par le programme `probA` [136]
2. la généralisation des alignements sous-optimaux en utilisant le serveur web `noptalign` qui implémente la méthode de Zuker [132].

3. les k-alignements prédits par notre algorithme `SubOpt`, où l'alignement initial utilisé est l'alignement global de Needleman-Wunsch.

Les deux premières méthodes ont été présentées dans la sous-section 5.2.3 à la page 93.

Dans [123], Sierk et ses collaborateurs ont construit un modèle de régression logistique pour améliorer la qualité d'alignement. Ce modèle cherche une meilleure combinaison des paramètres pour prédire la probabilité $P(a_i, b_j)$ que le nucléotide a_i soit aligné avec b_j dans les alignements réels. Ils remplacent ensuite la matrice de substitution par la table de probabilité P obtenue précédemment pour prédire les alignements. La qualité des alignements prédits a été bien améliorée.

Nous allons calculer les fréquences des paires de résidus et les entropies de position spécifique à partir des alignements sous-optimaux des trois méthodes : `SubOpt`, `noptalign`, `probA`.

5.4.3.1 Fréquence et entropie de position spécifique

Soient a et b les deux séquences d'acides d'aminés de taille n et m. Soit $\mathcal{A}$ un ensemble des alignements sous-optimaux de séquences a et b.

Définition 5.4 ***La fréquence du couple*** (a_i, b_k) *dans l'ensemble des alignements* $\mathcal{A}$ *est définie comme suit :*

$$\boldsymbol{f(a_i, b_k)} = \frac{N_{a_i,b_j}}{N_{\mathcal{A}}} \tag{5.5}$$

où a_i *est l'acide aminé à la position* i *de la séquence* a *et* b_k *est un acide aminé ou un gap de la séquence* b *:*

- *si* $k = 0$, b_k *est le gap avant le premier acide aminé de la séquence* b.
- *si* $k = 2, 4, 6, \ldots, 2m$, b_k *l'acide aminé à la position* $k/2$ *de la séquence* b.
- *si* $k = 3, 5, 7, \ldots, 2m\text{-}1$, b_k *est le gap entre les acides aminés à la position* $(k-1)/2$ *et* $(k+1)/2$ *de la séquence* b.
- *si* $k = 2m+1$, b_k *est le gap après le dernier acide aminé de la séquence* b.

$N_{a_i b_j}$ *est le nombre des alignements dans* $\mathcal{A}$ *où le nucléotide* a_i *est aligné avec le nucléotide ou le gap* b_k. $N_{\mathcal{A}}$ *est le nombre total des alignements dans l'ensemble* $\mathcal{A}$.

Définition 5.5 ***L'entropie de la position*** *i* ***de la séquence*** *a dans l'ensemble des alignements $\mathcal{A}$ est définie comme suit :*

$\forall 1 \leq i \leq n,$

$$\boldsymbol{H(a_i)} = -\sum_{k=0}^{2\cdot m} f(a_i, b_k) \cdot \ln f(a_i, b_k) \tag{5.6}$$

L'entropie $H_1(a_i)$ sera faible si l'acide aminé a_i s'aligne souvent avec l'acide aminé ou le gap à la même position dans $\mathcal{A}$. L'entropie $H_1(a_i)$ sera élevée, si l'acide aminé a_i s'aligne avec des acides aminés ou des gaps à différentes positions dans $\mathcal{A}$. L'entropie minimum est 0 et l'entropie maximum est égale à : $-\sum_{i=1}^{n} \frac{1}{2\cdot m+1} \cdot \ln(\frac{1}{2\cdot m+1}) = \ln(2 \cdot m + 1)$, où m est la longueur de la séquence b.

Nous calculons les entropies de position spécifique sur des alignements deux-à-deux sous-optimaux des protéines : $1ihvA$ (l'identifiant de Swiss Prot est $P00383$) et $1pht$ (l'identifiant de Swiss Prot est $P27986$). Ces deux protéines venant de *Escherichia coli* et de *Homo sapiens* contiennent un domaine $SH3$ qui est composé de cinq feuillets β [133].

Pour chacune des trois méthodes : `SubOpt`, `noptalign`, `probA`, nous générons 111 alignements sous-optimaux des séquences $1ihvA$ et $1pht$: $\mathcal{A}_{\texttt{SubOpt}}, \mathcal{A}_{\texttt{noptalign}}, \mathcal{A}_{\texttt{probA}}$, et calculons les entropies de position spécifique sur les acides aminés des séquences $1ihvA$ et $1pht$ comme suit :

1. Calculer les fréquences $f(1pht_i, 1ihvA_k)$ dans les ensembles des alignements $\mathcal{A}_{\texttt{SubOpt}}, \mathcal{A}_{\texttt{noptalign}}, \mathcal{A}_{\texttt{probA}}$.
2. Calculer les fréquences $f(1ihvA_i, 1pht_k)$ dans les ensembles des alignements $\mathcal{A}_{\texttt{SubOpt}}, \mathcal{A}_{\texttt{noptalign}}, \mathcal{A}_{\texttt{probA}}$.
3. Pour chaque acide aminé de la séquence $1ihvA$, calculer son entropie de position spécifique des programmes `SubOpt`, `noptalign`, ou `probA` :

$$H_{1hvA}(i) = -\sum_{k=0}^{2*m} f(1ihvA_i, 1pht_k) * \ln(f(1ihvA_i, 1pht_k)) \tag{5.7}$$

   Nous obtenons la figure 5.7.

4. Pour chaque acide aminé de la séquence $1pht$, calculer son entropie de position spécifique des programmes `SubOpt`, `noptalign`, ou `probA` :

$$H_{1pht}(i) = -\sum_{k=0}^{2*n} f(1pht_i, 1ihvA_k) * \ln(f(1pht_i, 1ihvA_k)) \tag{5.8}$$

Nous obtenons la figure 5.8.

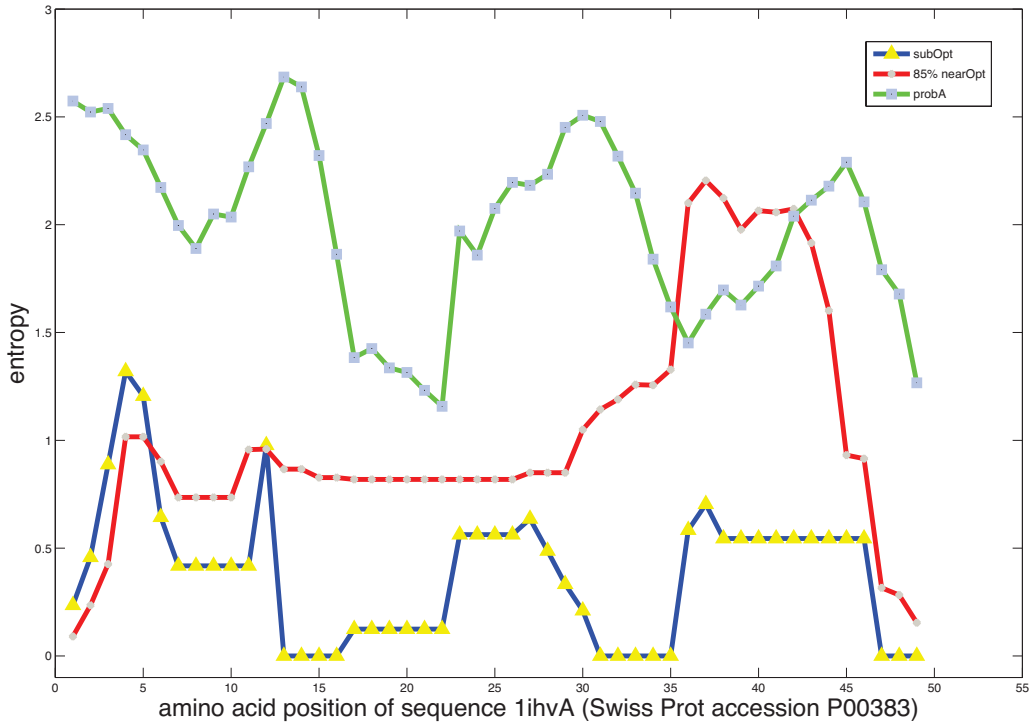


FIGURE 5.7 – Entropies de position spécifique de toute les positions de la séquence $1ihvA$ calculées à partir des alignments générés par des programmes `SubOpt`, `noptalign`, `probA`.

Nous nous demandons si un résidu à une position de l'une des deux séquences est aligné toujours avec un résidu à l'autre position de l'autre séquence parmi l'ensemble des alignements sous-optimaux. Si c'est le cas, pourquoi cette paire de résidus existe-t-elle dans l'ensemble des alignements sous-optimaux ?

Pour répondre à ces questions, nous regardons les figures 5.7 et 5.8. Nous observons que les entropies de position spécifique sortant de `SubOpt` semblent être plus petites que les deux autres méthodes, ce qui suggère deux raisons possibles : 1) il semble y avoir une plus grande diversité dans les alignements sous-optimaux générés par `noptalign` et `probA` que par `SubOpt`. 2) en regardant avec les régions des acides aminés fiablement alignés des séquences $1ihvA$ et $1pht$ extrait de *BAliBASE* (voir la figure 5.9), il semble que la région des acides aminés fiablement alignés corresponde approximativement à la région ayant les faibles entropies de position spécifique en particulier à l'égard de `SubOpt`. Dans la sous-section suivante, nous allons

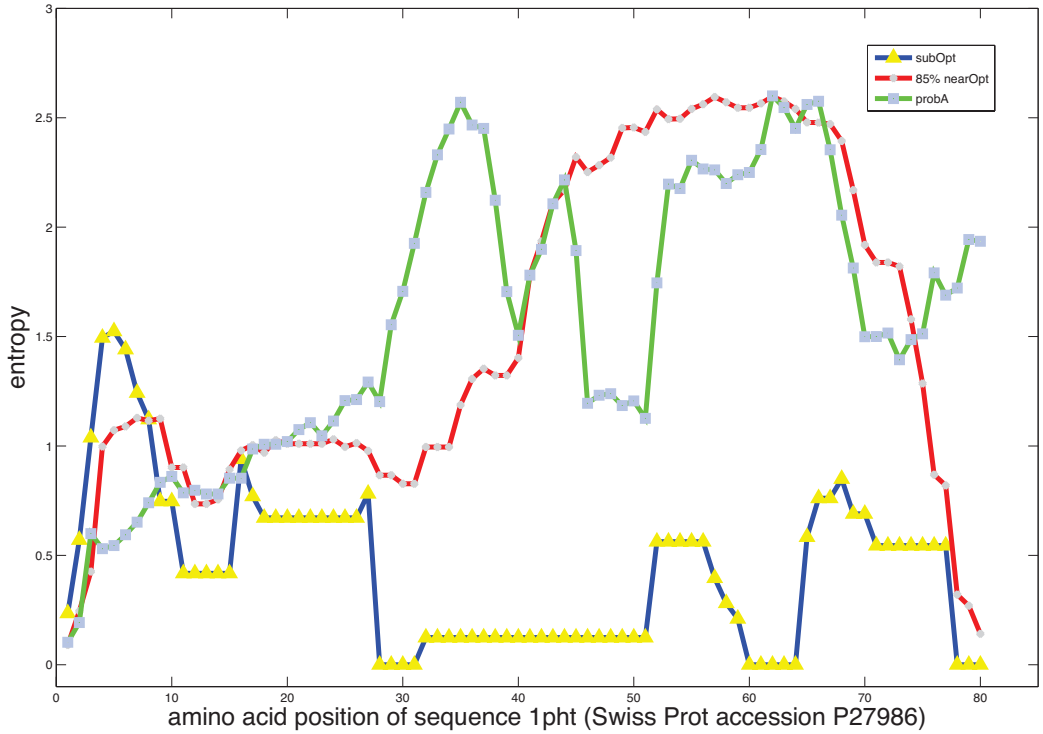


FIGURE 5.8 – Entropies de position spécifique de toute les positions de la séquence 1*pht* calculées à partir des alignments générés par des programmes `SubOpt`, `noptalign`, `probA`.

donc quantifier numériquement la diversité des alignements sous-optimaux et la corrélation entre la région des résidus fiablement alignés et la région ayant une faible entropie de position spécifique.

```
-NFRVYYRDsrd------pvwkGPAKLLWkg-----------------eGAVVIQd--nsdiKVVPRRKAKIIRd-----
gYQYRALYDykkereedidlhlGDILTVNkgslvalgfsdgqearpeeiGWLNGYnettgerGDFPGTYVEYIGrkkisp
```

FIGURE 5.9 – L'alignement de référence dans *BAliBASE* entre les deux séquences de protéines qui contiennent le domaine de SH3 : 1*ihvA* et 1*pht*. Les acides aminés en majuscules désignent les paires de résidus fiablement alignées.

5.4.3.2 Diversité sous-optimaux et corrélation avec la région fiablement alignée

Définition 5.6 ***La diversité de l'ensemble des alignements sous-optimaux*** *entre les deux séquences a et b est calculée comme suit :*

$$\boldsymbol{D(a,b)} = \sum_{i=1}^{n} \sum_{k=1}^{2m+1} f(a_i, b_k) \cdot (1 - f(a_i, b_k)) + \sum_{i=1}^{m} \sum_{k=1}^{2n+1} f(b_i, a_k) \cdot (1 - f(b_i, a_k)) \tag{5.9}$$

où la première somme est calculée sur toutes les fréquences des acides aminés de la séquence a, la deuxième somme est calculée sur toutes les fréquences des acides aminés de la séquence b. Cette définition de la diversité d'un ensemble d'alignements générés est inspirée de la diversité des structures secondaires proposée par le groupe de Vienna Package [64] (voir page 41). Plus la diversité est petite, plus les alignements sont analogues. Plus la diversité est grande, plus les alignements sont diverses.

Définition 5.7 ***Le coefficient de corréation de Pearson*** *[137] entre les positions des acides aminés fiablement alignées et les entropies de position spécifique sur les deux séquences* $1ihvA$ *et* $1pht$ *est calculé comme suit :*

$$r = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}} \tag{5.10}$$

où X_i est l'entropie de l'acide aminé à la position i de l'un des deux séquences. Y_i indique si l'acide aminé à la position i de l'un des deux séquences est fiablement aligné dans l'alignement de référence de *BAliBASE* : $Y_i = 1$, si c'est le cas, $Y(i) = 0$ sinon.

Les valeurs de r sont compris entre -1 et 1.

- $r = 0$ ou voisin de 0 signifie une absence de relation entre les deux distributions X et Y.
- $r < 0$: signifie une relation négative.
- $r > 0$: signifie une relation positive.
- $r = 1$ ou $r = -1$: signifie une relation linéaire parfaite positive ou négative.

Nous calculons d'abord le coefficient de corrélation de Pearson sur tous les acides aminés de la séquence $1ihvA$. Ce coefficient est noté comme **corrélation 1** dans la table 5.4. Nous calculons ensuite le cofficient sur toutes les acides aminés de la séquence $1pht$. Ce cofficient est

noté **corrélation 2** dans la table 5.4.

	SubOpt	noptalign	probA
diversité	30.65	74.4	85.72
corrélation 1	0.22	0.07	-0.11
corrélation 2	0.49	0.17	-0.14

TABLE 5.4 – Diversité d'alignements sous-optimaux et la corrélation de Pearson entre les entropies de position spécifique et la région fiablement alignée des trois méthodes différentes.

La table 5.4 nous montre une quantification numérique des deux idées mentionnées précédemment. 1) Il semble y avoir une plus grande diversité dans les alignements sous-optimaux engendrés par `noptalign` et `probA` que par `SubOpt`. En effet, la diversité de `probA` est 85.72, celle de `noptalign` est 74.4, et celle de `SubOpt` est 30.65. 2) La table 5.4 montre que la **corrélation 1** est 0.22 pour `SubOpt`, ce qui est beaucoup plus grand que les valeurs de 0.07 pour `noptalign` et -0.11 pour `probA`. D'une manière similaire, la **corrélation 2** est 0.49 pour `SubOpt`, qui est aussi beaucoup plus grande que les valeurs de 0.17 pour `noptalign` et -0.14 pour `probA`. Selon ces premières investigations, il apparaît que notre méthode `SubOpt` est plus efficace pour identifier la région biologiquement significative d'un alignement deux-à-deux des protéines.

5.5 Discussion

Dans ce chapitre, nous avons présenté un nouvel algorithme `SubOpt` pour l'analyse des séquences de protéines, qui est motivé par l'intérêt de l'amélioration de la qualité d'alignement de séquences deux-à-deux. Les alignements mathématiquement optimaux de séquences, produits par l'application de programmation dynamique avec des matrices de substitution (BLOSUM, PAM, etc) n'alignent pas toujours correctement les paires de résidus du site fonctionnel ou des éléments structuraux. En effet, depuis longtemps, nous considérons que l'alignement de séquences est moins précis que l'alignement structurel, en comparant avec les alignements de référence manuellement définies. Cependant, avec la croissance exponentielle de bases de données de séquences protéiques, il reste un domaine important de recherche en bioinformatique pour améliorer la qualité des alignements de séquences deux-à-deux et multiples.

Nous avons en même temps présenté une nouvelle méthode de génération d'alignements deux-à-deux sous-optimaux globaux et locaux des séquences de protéines. Étant donné n'import quel alignement initial $\mathbb{A}_0$ de deux séquences d'acides aminés, en temps cubique, notre algorithme `SubOpt` calcule, pour toutes les valeurs de k, l'ensemble des k-alignements. En utilisant les alignements de référence de *BAliBASE*, nous avons comparé notre algorithme `SubOpt` avec l'algorithme de Needleman-Wunsch, et montré que pour la plupart, les alignements prédits par notre programme sont plus proches des alignements de référence de *BAliBASE* que ne le sont ceux prédits par l'alignement optimal de Needleman-Wunsch. De plus, nous avons calculé la diversité et les entropies de position spécifique des alignements sous-optimaux produits par `SubOpt`, `probA` et `noptalign`. Nous avons observé que les alignements engendrés par `SubOpt` sont moins divers, dans le sens que les entropies de position spécifique sont plus corrélées avec les positions des paires de résidus fiablement alignées selon *BAliBASE*. Pour cette raison, il est possible que nos fréquences de paires de résidus puissent conduire à des améliorations futures dans l'alignement de séquences, par exemple, en intégrant les fonctionnalités de notre méthode dans le modèle de Sierk[123]. Une autre future orientation possible est de calculer les alignements multiples sous-optimaux de séquences en étendant la méthode deux-à-deux présentée ici.

Chapitre 6

Conclusion

Nous avons d'abord présenté un nouvel algorithme qui permet d'estimer la densité relative d'états d'énergie des structures secondaires d'une séquence ou d'une hybridation de deux séquences d'ARN et permet de calculer la température de dénaturation d'une hybridation de deux molécules d'ARN. Le programme que nous avons implémenté prédit beaucoup plus rapidement la densité d'états d'énergie que le programme `RNAsubopt`. Pour la plupart des séquences testées, le pipeline que nous avons construit prédit mieux la température de dénaturation que les deux autres programmes existants. Cependant, le vrai avantage de notre algorithme est qu'il n'y a pas de restriction sur les interactions autorisées. Contrairement aux approches de programmation dynamique, toutes les structures secondaires et hybridations peuvent être générées par notre échantillonnage. Si nous avons un modèle d'énergie pour les structures secondaires avec pseudo-noeuds, le problème NP-complet de la prédiction des quantités thermodynamiques des structures secondaires avec pseudo-noeuds peut être résolu approximativement par notre algorithme.

Nous avons ensuite présenté un nouvel outil qui recherche l'ensemble de structures sous-optimales d'une séquence et d'une structure d'ARN donnée, nous calculons en même temps la pseudo-fonction de partition de Boltzmann pour ses structures k-voisines. En comparant avec cinq autres programmes existants, nous avons observé que la plupart des structures prédites par notre programme sont les structures le plus similaires structurellement aux structures fonctionnelles attendues de riboswitchs.

Nous avons enfin présenté un nouvel algorithme d'alignement global et local pour l'analyse des séquences de protéines. En utilisant les alignements de référence de *BAliBASE*, nous avons

comparé notre algorithme avec l'algorithme de Needleman-Wunsch, et montré que pour la plupart, les alignements prédits par notre programme sont plus proches des alignements de référence de *BAliBASE* que ne le sont ceux prédits par l'alignement optimal de Needleman-Wunsch. De plus, Nous avons observé que nos alignements sont moins divers, dans le sens que les entropies de position spécifique sont plus corrélées avec les positions des paires de résidus fiablement alignées selon *BAliBASE*. Pour cette raison, il est possible que nos fréquences de paires de résidus puissent conduire à des améliorations futures dans l'alignement de séquences, par exemple, en intégrant les fonctionnalités de notre méthode dans le modèle de régression logistique.

Chapitre 7

Annexe

Les temps d'exécution du program RNA-WL et du programme RNAsubopt

Voici les données pour tracer la figure 3.10.

– **Longueur 30-40 :**

>X00063.1/1061-1096 : CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUU
>L00073.31/390-426 : CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUUC
>D86625.1/6-35 : GUUCUUGUUUCAACAGUGAUUGAACGGAAC
>BC011157.1/1366-1401 : UGCUUUAAGGAAAAACCGAAUAAAGAUUUCAUGUUU
>M26901.1/338-374 : CUGCUUUGAGGACAAAGAGAAUAAAGACUUCAUGUUC

ID	Longueur	RNA-WL	RNAsubopt	Test de χ^2	value-P
X00063.1/1061-1096	36	637.81	48.96	0.0252	0
L00073.31/390-426	37	566.41	70.39	0.0280	0
D86625.1/6-35	30	411.33	4.43	0	0
BC011157.1/1366-1401	36	141.09	43.03	0	0
M26901.1/338-374	37	144.88	53.78	0	0
mean	35.2	380.30	44.12	–	–

– **Longueur 40-45 :**

>AB010982.1/1-45 : AUGAACAACCAACGAAAAAGGACGGGAAAACCGUCUAUCAAUAUG
>AF022216.1/537-579 : GGGUAAGAGGUUCUAGCUACCCUCUCAAAAAAACUAAGGAGAA
>U32798.1/8833-8789 : CCCCCCGUAGUUCGCAAACCUCCUACAAUAAAAAACUAGGUAAAA
>AL935260.1/111560-111516 : GCUGGCCUGGUUCGUAAACUUCCCAGGAUAAAAAACCAAGAACUU
>AB010990.1/1-45 : AUGAACAACCAACGAAAAAAGACGGGAAGACCGUCUAUCAAUAUG

ID	Longueur	`RNA-WL`	`RNAsubopt`	Test de χ^2	value-P
AB010982.1/1-45	45	657.89	179.36	0.0743	0
AF022216.1/537-579	43	56.23	303.99	0	0
U32798.1/8833-8789	45	706.64	211.77	0	0
AL935260.1/111560-111516	45	761.59	2002.63	0	0
AB010990.1/1-45	45	501.70	110.29	0	0
mean	44.6	536.81	561.608	–	–

– **Longueur 45-50 :**

>AL162754.2/2823-2868 : AACUUCGCGGUUCGAAAACCUCCCGCGUCACCAAAACUAGGAUUCG
>BX571857.1/755298-755252 : AUAUCAGAGGUUCCUAGCUGAAACCCUCUAUAAAAAACUAGACAUUG
>AE007742.1/1685-1638 : UUUAAGACAGUUCGAAACCAUCCUGUCUAUAAAUAAAACUAUGGAGGU
>AE015944.1/68502-68549 : AUAUGGACAGUUCGUAACCAUCCUGUCCCUAAAUAAAACUAUGGAGGU
>AF036741.1/118-165 : AUCCAGCCGACGAGUCCCAAAUAAAACGAAACGCGCGUCAAAGUGGAU

ID	Longueur	RNA-WL	RNAsubopt	Test de χ^2	value-P
AL162754.2/2823-2868	46	1207.24	3530.60	0	0
BX571857.1/755298-755252	47	4051.16	12838.27	0	0
AE007742.1/1685-1638	48	718.27	42619.88	0	0
AE015944.1/68502-68549	48	215.31	1459.75	0	0
AF036741.1/118-165	48	1134.94	30526.03	0	0
mean	47.4	1465.384	18194.906	–	–

– **Longueur 50-55 :**

>K02120.1/628-682 : AUGGGAAAUUCCCCCUCCUAUAACCCCCCCGCUGGUAUCUCCCCCUCAGACUGGC
>BA000007.2/1165782-1165731 : UUCUGGUGACAUUUGGCGGUAUCAGUUUUACUCCGUAACUGCUCUGCCGCCC
>M21212.1/157-106 : CAACAGCGAAGCGGAACGGCGAAACACACCUUGUGUGGUAUAUUACCCGUUG
>M17439.1/226-177 : AAACAGAGAAGUCAACCAGAGAAACACACGUUGUGGUAUAUUACCUGGUA
>AF370716.1/3656-3603 : UGUGUGUAGUACUUGGCGGCAUCAGUUUUUCUUAGUCCUUUCUGAUGUCCGCCC

ID	Longueur	RNA-WL	RNAsubopt	Test de χ^2	value-P
K02120.1/628-682	55	3028.42	56878.90	0	0
BA000007.2/1165782-1165731	52	2019.05	327470.33	0	0
M21212.1/157-106	52	5994.45	0	0	0
M17439.1/226-177	50	2386.84	0	0	0
AF370716.1/3656-3603	54	1297.96	0	0	0
mean	53.5	2524.035	192174.615	–	–

– **Longueur 55-60 :**

>BC056833.1/241-299 : UGAUGCCCCUCACCCACCUCUGAAGAUCCCAGGUGGGCGAGGGAACGGAGCACGGGAUC
>AF480891.2/120-178 : CCACCUUAAGGCCGCGCUCGCCAGCCUCGGCGGGGCGGCUCCCGCCGCCGCAACCAAUG
>AB003688.1/2994-3052 : CCACCUUAAGGCCGCGCUCGCCAGCCUCGGCGGGGCGGCUCCCGCCGCCGCAACCAAUG

>AF306514.1/49-107 : GUGGAGUCAGGCCAGCAAAAGCUGCCACCGGAUACUGAGUA-GACGGUGCUGCCUGGGUU
>D10706.1/256-314 : UGAUGUCCCUCACCCACCCCUGAAGAUCCCAGGUGGGCGAGG-GAACAGUCAGCGGGAUC

ID	Longueur	`RNA-WL`	`RNAsubopt`	Test de χ^2	value-P
BC056833.1/241-299	59	1928.97	345455.04	0	0
AF480891.2/120-178	59	8814.08	∞	–	–
AB003688.1/2994-3052	59	8819.28	∞	–	–
AF306514.1/49-107	59	3798.01	∞	–	–
D10706.1/256-314	59	2697.79	∞	–	–
mean	59	5211.63	345455.04	0	0

the sequence BC056833.1/241-299 is tested during a moment of 25 days, and the other sequences are tested during a smaller moment.

– **Longueur 60-65 :**

>X96641.1/1-62 : GGCUAAUGAUGGAAAAAUCAUUAUUGGAAAAGAAUGACAUGAA-CAAAGGAACCACUGAAGUG
>AE006052.1/10631-10566 : UUCAUAGUGGUUCGUAACCCUCCCACUUGAACAACCAA-CAAUUGUUCGAAACAAAAC
>U02551.1/12-76 : ACUCUUUAGCGUUGGACGGUUACGUCUAGUCGGGUGAUUAGC-CAGACUCUAACUUAUUGAACGUA
>X66717.1/2272-2334 : CUACUCUUGUACAGAAUGGUAAGCACGUGUAAUAGGAG-GUACAAGCCACCCUAUUGCAUAUUA
>AF207902.1/31022-31083 : ACACUCUCUAUCAGAAUGGAUGUCUUGCUGUCAUAACA-GAUAGAGAAGGUUGUGGCAGACCC

ID	Longueur	`RNA-WL`	`RNAsubopt`	Test de χ^2	value-P
X96641.1/1-62	62	33996.26	∞	–	–
AE006052.1/10631-10566	65	11952.53	∞	–	–
U02551.1/12-76	65	5784.73	∞	–	–
X66717.1/2272-2334	63	5130.03	∞	–	–
AF207902.1/31022-31083	62	3533.34	∞	–	–
mean	63.4	12079.38	∞	–	–

– **Longueur 65-70 :**

>AF427793.1/1040-1105 : UGACAACGGCGAAGGCCGAGCCUAGCAACCCGGGCGGCG-GAUCGCCGUCCUUGCAACAAGCUCGUU
>L22531.1/397-465 : CAAAGUAAUUUUCGUGCUCUCAACAAUUGUCGCCGUCACA-GAUUGUUGUUCGAGCCGAAUCUUACUUCU
>AF261825.2/44164-44233 : ACUCUUUAGCGUUAGGCUUUGAUUUAUAGCCUUGUC-GAGCGUUUCGCCAGACACUAACUUAUUGAGUACU
>AB079134.1/692-757 : GCUCUUUAGCUUAGGACGAAUUUCGUCUAGUCGGGU-GAUUAGCCAGACUCUAACUUAUUGAACGGG
>AF078527.1/2546-2614 : UCGCGCAUCUUGUUGUCCAAGUGUAGUUUUUGGCGAAAC-CAUUUGAUCAUGCAACAAGAUGCGCUUCCA

ID	Longueur	`RNA-WL`	`RNAsubopt`	Test de χ^2	value-P
AF427793.1/1040-1105	66	42362.92	∞	–	–
L22531.1/397-465	69	10849.84	∞	–	–
AF261825.2/44164-44233	70	5263.96	∞	–	–
AB079134.1/692-757	66	7902.46	∞	–	–
AF078527.1/2546-2614	69	3677.76	∞	–	–
mean	68	14011.39	∞	–	–

– **Longueur 70-75 :**

>AJ5325.1/3-77 : AGCAAAAUGAGGAGGUUAUAUGAUCUGAAGUAUCUAAUGUU-GAUAAUUCAAGGUCUUCAAAGUUCUCUGAUUGCU
>AE004309.1/1945-2015 : GCUCUUUAGCGAAUGGACUACACAUAUUAGUCUA-GUCGGGCGUUUAGCCAGACUGAAAUUUAUUGAACGAC
>Z12832.1/542-622 : AAACUUUUUCGAUAGCAUUAGCUGGUUACAGGUCUGGCGUGU-GUUUCGCCAGGCCCAGUUGCCACGUAAUGUG
>AE013691.1/12531-12457 : UGAAAGACGCGCAUUUGUUAUCAUCAUCCCUUCUAUUA-GAGAUGUUAAUUUGGCCACAGUGAUGUGGCCUUUUCU
>X60206.1/2132-2205 : CACCCUUAGCGAGAGGUUUAUCAUUAAGGUCAACCUCUG-GAUGUUGUUUCGGCAUCCUGCAUUGAAUCUGAGUU

ID	Longueur	`RNA-WL`	`RNAsubopt`	Test de χ^2	value-P
AJ5325.1/3-77	75	4375.74	∞	–	–
AE004309.1/1945-2015	71	21439.61	∞	–	–
Z12832.1/542-622	73	117195.59	∞	–	–
AE013691.1/12531-12457	75	34955.23	∞	–	–
X60206.1/2132-2205	74	14978.02	∞	–	–
mean	73.6	38588.838	∞	–	–

Les scores de NestedAlign

Voici les données pour tracer la figure 4.11.

index	EMBL	RNAbor	RNAborMEA	sampleRNAbor	RNAlocopt	RNAshapes	UNAFold
0	AL591981/205922-205823	-9.0	14.0	-9.0	-8.5	-9.0	-9.0
1	CP000764/271074-271175	-43.5	13.0	-37.5	-44.5	-23.0	-53.0
2	CP000764/308099-308200	-27.0	-12.5	-24.5	-31.5	-25.5	-22.0
3	BA000028/760473-760574	-25.5	2.5	-36.0	-38.5	-24.5	-31.0
4	CP000557/252200-252301	-9.5	18.0	-9.5	8.5	-10.0	-12.0
5	X83878/168-267	60.0	101.5	57.0	66.0	64.0	59.0
6	BA000004/1593074-1592973	35.0	17.5	-13.5	-21.5	-19.0	-13.5
7	AAOX01000023/19446-19345	-15.0	8.0	-13.0	-18.5	-13.5	-15.5
8	CP000416/1798040-1798138	5.5	5.5	1.5	12.0	4.5	-4.5
9	CP000721/398929-399026	26.0	29.5	16.5	-20.0	21.5	-32.0
10	BA000028/1103943-1104044	1.0	9.5	2.0	-0.5	0.5	0.5
11	ABDQ01000002/251055-251152	-16.0	4.5	-16.5	-21.5	-17.5	-22.5
12	AAXV01000026/31334-31233	11.5	10.5	-1.5	-8.5	22.0	-3.0
13	AE016877/298774-298875	-18.5	21.0	-17.5	-34.0	-12.0	-26.5
14	BA000004/676475-676576	-28.5	-22.5	-28.0	-69.0	-21.0	-29.5
15	AE017333/692981-693082	-1.5	8.5	-11.5	-9.5	-5.5	-53.0
16	AM180355/256217-256318	-17.0	-38.5	-45.5	-49.0	-48.0	-49.0
17	AM406671/1321062-1320965	-25.5	-2.0	-22.0	-28.5	-23.5	-23.5
18	CP000612/2598111-2598012	-42.0	-32.5	-42.0	-47.5	-39.0	-38.5
19	CP000002/697032-697134	-8.0	-7.0	-10.5	-10.0	-4.5	-7.5
20	CP000002/2295936-2295837	23.5	47.0	31.5	21.0	30.0	22.5
21	AL596170/223345-223246	-0.5	10.0	0.5	-8.5	-10.0	-10.0
22	ABDQ01000005/131908-131807	-33.0	-12.0	-31.5	-31.5	-19.0	-50.0
23	AAOX01000052/9069-8968	-13.5	7.5	-14.0	-21.0	-15.5	-14.5
24	AE017333/4024324-4024425	-29.5	-20.5	-33.5	-24.0	-23.5	-36.0
25	AP006627/1554717-1554818	-31.5	4.5	-37.0	-44.5	-28.5	-43.5
26	CP000024/1182948-1183043	-0.5	-12.5	-9.0	4.0	2.0	-19.0
27	BA000028/786767-786867	-18.0	-36.5	-48.0	-46.5	-49.0	-44.5
28	ABDP01000002/29688-29587	-34.5	-37.5	-34.5	-37.0	-35.0	-50.0
29	BA000043/272473-272574	-9.5	11.5	-9.5	-10.0	-3.0	-12.5
30	CP000724/944285-944386	-30.5	-16.5	-30.5	-28.5	-26.5	-31.5
31	CP000764/1409725-1409826	14.0	6.5	-18.0	-24.0	-11.5	-20.0
32	AAEK01000017/86437-86538	-44.5	-38.0	-41.5	-52.0	-35.0	-49.0
33	CP000764/357645-357544	11.0	-8.0	-33.0	-26.0	-18.5	-36.0

Voici les données pour tracer la figure 4.12.

index	EMBL	RNAbor	RNAborMEA	sampleRNAbor	RNAlocopt	RNAshapes	UNAFold
0	AL591981/205922-205823	27.5	44.0	28.5	25.5	25.5	25.5
1	CP000764/271074-271175	13.0	17.0	11.0	6.5	12.0	5.5
2	CP000764/308099-308200	24.0	25.0	26.5	23.0	24.5	26.5
3	BA000028/760473-760574	18.5	23.0	13.0	20.5	23.5	23.0
4	CP000557/252200-252301	7.0	12.0	7.0	10.0	6.5	4.5
5	X83878/168-267	143.0 1	55.5 1	43.0	141.0	143.0	141.0
6	BA000004/1593074-1592973	41.0	40.0	41.0	36.0	38.0	41.0
7	AAOX01000023/19446-19345	47.5	51.0	46.0	42.5	34.0	43.5
8	CP000416/1798040-1798138	17.5	14.0	12.5	13.0	11.5	12.5
9	CP000721/398929-399026	36.5	27.5	23.0	-38.5	34.5	-52.5
10	BA000028/1103943-1104044	32.0	30.5	32.0	27.5	30.5	30.0
11	ABDQ01000002/251055-251152	27.0	23.0	26.5	24.0	25.5	7.5
12	AAXV01000026/31334-31233	37.5	41.5	38.0	32.5	35.0	36.0
13	AE016877/298774-298875	24.0	32.5	23.0	19.0	23.0	22.5
14	BA000004/676475-676576	9.0	-5.0	6.5	-35.5	5.0	9.0
15	AE017333/692981-693082	-30.0	-4.0	-23.5	-25.5	-17.0	-70.5
16	AM180355/256217-256318	-23.5	-21.0	-25.0	-27.0	-23.5	-27.0
17	AM406671/1321062-1320965	-0.5	12.5	1.0	-10.0	1.0	0.5
18	CP000612/2598111-2598012	-12.0	-1.0	-8.0	-8.5	-9.5	-9.0
19	CP000002/697032-697134	16.5	11.0	12.0	14.0	16.5	7.5
20	CP000002/2295936-2295837	75.0	78.5	75.5	71.0	72.0	69.5
21	AL596170/223345-223246	30.5	38.0	30.5	28.5	29.5	29.5
22	ABDQ01000005/131908-131807	12.5	-2.5	13.0	10.5	13.5	4.5
23	AAOX01000052/9069-8968	12.5	14.0	13.5	11.0	12.0	12.0
24	AE017333/4024324-4024425	-3.5	7.5	3.5	6.0	-2.5	-1.5
25	AP006627/1554717-1554818	22.5	7.5	22.5	14.5	25.5	12.5
26	CP000024/1182948-1183043	6.0	14.5	6.5	6.0	5.0	6.0
27	BA000028/786767-786867	-23.5	-11.0	-23.0	-24.5	-21.0	-24.0
28	ABDP01000002/29688-29587	3.0	1.5	2.5	1.0	4.5	0.5
29	BA000043/272473-272574	17.5	16.5	12.5	13.5	12.5	11.5
30	CP000724/944285-944386	10.0	19.0	10.5	7.0	12.0	9.5
31	CP000764/1409725-1409826	32.5	41.5	32.0	26.5	35.0	30.5
32	AAEK01000017/86437-86538	11.5	19.5	13.0	8.0	13.0	11.0
33	CP000764/357645-357544	23.5	29.5	24.5	24.0	22.0	22.5

Bibliographie

[1] Francis Crick. Central Dogma of Molecular Biology. *Nature*, 227 :561563, August 1970.

[2] Bock A, Forchhammer K, Heider J, Baron C. Selenoprotein synthesis :An expansion of the genetic code. *Trends Biochem. Sci.*, 16, 463467, 1991.

[3] Lim LP, Glasner ME, Yekta S, Burge CB, Bartel DP. Vertebrate microRNA genes. *Science*, 299, 1540, 2003.

[4] Mandal M, Boese B, Barrick J, Winkler W, Breaker R Riboswitches control fundamental biochemical pathways in Bacillus subtilis and other bacteria. *Cell*, 113(5) :577-586, 2003

[5] Ming T. Cheah, Andreas Wachter, Narasimhan Sudarsan and Ronald R. Breaker Control of alternative RNA splicing and gene expression by eukaryotic riboswitches. *Nature*, 447, 497500, 2007.

[6] Lyngso,R.B. and Pedersen,C.N. RNA pseudoknot prediction in energy-based models. *J. Comput. Biol.*, 7, 409427, 2000.

[7] Nawrocki EP, Kolbe DL, Eddy SR Infernal 1.0 : inference of RNA alignments. *Bioinformatics*, 25(10) :1335-1337, 2009.

[8] Knudsen,B. and Hein,J. Pfold : RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31, 34233428 2003.

[9] Zuker M, Stiegler P. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, 10 ;9(1) :133-48, 1981.

[10] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, 125,167188, 1995

[11] David H. Mathews, Matthew D. Disney, Jessica L. Childs, Susan J. Schroeder, Michael Zuker, and Douglas H. Turner Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure *PNAS*, May 11, 2004 vol. 101 no. 19 7287-7292

[12] N.R. Markham and M.Zuker. UNAFold : software for nucleic acid folding and hybridization. *Methods Mol. Biol.*, 453 :3–31, 2008.

[13] Stephan H Bernhart, Hakim Tafer, Ulrike Mckstein, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker Partition function and base pairing probabilities of RNA heterodimers. *Mol. Biol.*, 1, 3, 2006.

[14] Ulrike Muckstein, Hakim Tafer, Jorg Hackermuller, Stephan H. Bernhart, Peter F. Stadler and Ivo L. Hofacker Thermodynamics of RNA-RNA binding. *Bioinformatics*, 22, 11771182, 2006.

[15] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3) :443453, March 1970.

[16] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147(1) :195197, March 1981.

[17] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. Journal of Molecular Biology, 215(3) :403410, October 1990.

[18] S. F. Altschul and E. V. Koonin. Iterated profile searches with PSI-BLASTa tool for discovery in protein databases. *Trends Biochem. Sci.*, 23(11) :444447, November 1998.

[19] S. F. Altschul, T. L.Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic. Acids. Res.*, 25(17) :33893402, September 1997.

[20] Huanwang Yang, Fabrice Jossinet, Neocles Leontis, Li Chen, John Westbrook, Helen Berman and Eric Westhof Tools for the automatic identification and classification of RNA base pairs *Nucl. Acids Res.*, 31 (13) :3450-3460, 2003.

[21] Lemieux S, Major F. RNA canonical and non-canonical base pairing types : a recognition method and complete repertoire. *Nucl. Acids Res.*, 30(19) :4250-63, 2002.

[22] B.P. Lewis, C.B. Burge, and D.P. Bartel. Conserved seed pairing, often anked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120(1) :15 20, 2005.

[23] Victor Ambros, Bonnie Bartel, David P. Bartel, Christopher B. Burge, James C. Carrington, Xuemei Chen, Gideon Dreyfuss, Sean R. Eddy, Sam Griffiths-Jones, Mhairi Marshall, Marjori Matzke, Gary Ruvkun and Thomas Tuschl A uniform system for microRNA annotation *RNA*, 9 : 277-279, 2003.

[24] Ivey KN, Srivastava D MicroRNAs as regulators of differentiation and cell fate decisions *Cell Stem Cell*, 7 :3641, 2010.

[25] David Baulcombe An RNA Microcosm *Science*, 297 : 2002.

[26] S. Sassen, E.A. Miska, and C. Caldas. MicroRNA : implications for cancer. *Vir-chows Archiv*, 452(1) :1 10, 2008.

[27] J. Johansson, P. Mandin, A. Renzoni, C. Chiaruttini, M. Springer, and P. Cossart. An RNA thermosensor controls expression of virulence genes in Listeria monocytogenes. *Cell*, 110(5) :551 561, 2002.

[28] W. Winkler, A. Nahvi, and R.R. Breaker. Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. *Nature*, 419(6910) :952 956,2002.

[29] Parisien M, Major F. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*. 452(7183) :51-5, 2008.

[30] P. Brion and E. Westhof. Hierarchy and dynamics of RNA folding. Annual review of biophysics and biomolecular structure, 26(1) :113-137, 1997.

[31] J. Gorodkin, L. J. Heyer and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences *Nucl. Acids Res.*, 25 (18) : 3724-3732, 1997

[32] Mathews, D.H. and Turner, D.H. Dynalign : an algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, Volume 317, Issue 2, Pages 191-203, 22 March 2002.

[33] Sean R.Eddy and Richard Durbin RNA sequence analysis using covariance models *Nucleic Acids Research*, 1994, Vol. 22, No. 11 2079-2088

[34] R.Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proceedings of the National Academy of Sciences, USA*, 77(11) :6309–6313, 1980.

[35] Xia T, SantaLucia J Jr, Burkard ME, Kierzek R, Schroeder SJ, Jiao X, Cox C, and Turner DH. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry.* 37(42) :14719-35, 1998.

[36] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5) :911-940, 1999.

[37] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res. 31 (13), 3406-3415, 2003.

[38] Ivo L. Hofacker Vienna RNA secondary structure server *Nucl. Acids Res.* 31 (13) : 3429-3431, 2003

[39] Cédric Saule, Mireille Régnier, Jean-Marc Steyaert, and Alain Denise. Counting RNA Pseudoknotted Structures *Journal of Computational Biology.* October 2011, 18(10) : 1339-1351. doi :10.1089/cmb.2010.0086.

[40] Y. Ding, C. Y. Chan, and C. E. Lawrence. RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA*, 2005.

[41] H.Chitsaz, R.Salari, S.C. Sahinalp, and R.Backofen. A partition function algorithm for interacting nucleic acid strands. *Bioinformatics*, 25(12) :i365–i373, 2009.

[42] Nicholas Metropolis and Stanislas Ulam. The Monte Carlo Method *Journal of the American Statistical Association*, vol. 44, no 247, p. 335-341, septembre 1949.

[43] Nicholas Metropolis. The Beginning of the Monte Carlo Method. *Los Alamos Science*, 15, 125-130, 1987.

[44] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. *J Chem Phys*,21, 1087-1092, 1953.

[45] Binder K. Monte Carlo simulation in statistical physics. *Berlin, New York : Springer-Verlag*, p127, 1988.

[46] Binder K. Monte Carlo and molecular dynamics simulations in polymer sciences. *Oxford : Oxford University Press*, p587, 1995.

[47] Landau DP, Binder K. A guide to Monte Carlo simulations in statistical. *physics. New York : Cambridge University Press*, p384, 2000.

[48] Kalos MH. Monte Carlo methods. *New York : J. Wiley Sons*, 1986.

[49] Hastings, W.K. Monte Carlo Sampling Methods Using Markov Chains and Their Applications *Biometrika*, 97-109, 1970

[50] Swendsen, R. H., Wang, J. Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.*, 58(2) :868, 1987.

[51] Wang,F. and Landau,D.P. Determining the density of states for classical statistical models : a random walk algorithm to produce a flat histogram. *Phys. Rev. E*, 64, 056101(1)056101(16), 2001

[52] Wang,F. and Landau,D.P. Efficient, multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Lett.*, 86, 20502053, 2001

[53] R. E. Belardinelli and S. Manzi and V. D. Pereyra Analysis of the convergence of the 1t and Wang-Landau algorithms in the calculation of multidimensional integrals *Phys. Rev. E (American Physical Society)* 78 : 067701, 2008

[54] P. Ojeda and M. Garcia and A. Londono and N.Y. Chen Monte Carlo Simulations of Proteins in Cages : Influence of Confinement on the Stability of Intermediate States. *Biophys. Jour. (Biophysical Society)*, 96 (3) : 10761082, 2009

[55] Philip Bradley, Kira M. S. Misura and David Baker Toward high-resolution de novo structure prediction for small proteins. *Science*, 309, 18681871, 2005

[56] Rhiju Das and David Baker Automated de novo prediction of native-like RNA tertiary structures. *Proc. Natl Acad.* Sci. USA, 104, 1466414669, 2007

[57] Ortiz, AR, Kolinski A, and Skolnick J Fold assembly of small proteins using Monte Carlo simulations driven by restraints derived from multiple sequence alignments. *J. Mol. Biol.*, 277, 419448, 1998

[58] Christoph Flamm, Walter Fontana, Ivo L. Hofacker and Peter Schuster RNA folding at elementary step resolution. *RNA* 6, 325338, 2000

[59] Danilova LV, Pervouchine DD, Favorov AV, and Mironov AA. RNAKinetics : a web server that models secondary structure kinetics of an elongating RNA. *J. Bioinform. Comput. Biol.*, 4, 589596, 2006

[60] Isambert,H. and Siggia,E.D. Modeling RNA folding paths with pseudoknots : application to hepatitis delta virus ribozyme. *Proc. Natl Acad. Sci. USA*, 97, 65156520, 2000

[61] A. Xayaphoummine, T. Bucher, and H. Isambert Kinefold web server for RNA/DNA folding pathand structure prediction including pseudoknots and knots. *Nucleic Acids Res.*, 33, W605W610, 2005

[62] Jan Cupal , Ivo L. Hofacker , Peter F. Stadler Dynamic programming algorithm for the density of states of RNA secondary structures. *Computer Science and Biology 96 (Prooceedings of the German Conference on Bioinformatics)*, Univ. Leipzig. pp. 184-186, 1996.

[63] Stefan Wuchty Walter Fontana Ivo L. Hofacker Peter Schuster Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49, 145164, 1999.

[64] Gruber AR, Lorenz R, Bernhart SH, Neuböck R, Hofacker IL. The Vienna RNA websuite. *Nucleic Acids Res.* 2008 Jul 1 ;36. Epub 2008 Apr 19.

[65] Steven R Morgan and Paul G Higgs Barrier heights between ground states in a model of RNA secondary structure *J. Phys. A : Math. Gen.* 31 3153 doi : 10.1088/0305-4470/31/14/005, 1998

[66] P.Clote, E.Kranakis, D.Krizanc, and B.Salvy. Asymptotics of canonical and saturated RNA secondary structures. *J. Bioinform. Comput. Biol.*, 7(5) :869–893, October 2009.

[67] R.A. Dimitrov and M.Zuker. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal*, 87 :215–226, 2004.

[68] N.R. Markham. Algorithms and software for nucleic acid sequences, 2006.

[69] Voss B, Meyer C, Giegerich R. Evaluating the predictability of conformational switching in RNA. *Bioinformatics* 20(10) :1573-82, Jul 10, 2004.

[70] Steffen P, Voss B, Rehmsmeier M, Reeder J, Giegerich R RNAshapes : an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4) :500503, 2006.

[71] Freyhult E, Moulton V, Clote P Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics*, 23(16) :20542062, 2007.

[72] Zhi John Lu, Jason W. Gloor, and David H. Mathews Improved RNA secondary structure prediction by maximizing expected pair accuracy *RNA*, 15 : 1805-1813, 2009.

[73] M Zuker On finding all suboptimal foldings of an RNA molecule *Science* 7 Vol. 244 no. 4900 pp. 48-52, April 1989.

[74] Ye Ding and Charles E. Lawrence A statistical sampling algorithm for RNA secondary structure prediction *Nucl. Acids Res.* 31 (24) : 7280-7301, 2003.

[75] S. Wuchty, W. Fontana, I. L. Hofacker and P. Schuster Complete Suboptimal Folding of RNA and the Stability of Secondary Structures *Biopolymers*, 49, 145-165, 1999.

[76] E. Freyhult, V. Moulton, P. Clote. RNAbor : A web server for RNA structural neighbors. *Nucleic Acids Res.* Jul 1 ;35(Web Server issue) :W305-9. Epub 2007 May 25.

[77] Olsthoorn R, Mertens S, Brederode F, Bol J A conformational switch at the 30 end of a plant virus RNA regulates viral replication. EMBO J. 1999, 18 :48564864.

[78] Repsilber D, Wiese S, Rachen M, Schroder A, Riesner D, Steger G Formation of metastable RNA structures by sequential folding during transcription : time-resolved structural analysis of potato spindle tuber viroid-stranded RNA by temperature-gradient gel. RNA 1999, 5 :574584.

[79] Ray PS, Jia J, Yao P, Majumder M, Hatzoglou M, Fox PL A stress-responsive RNA switch regulates VEGFA expression. Nature 2009, 457(7231) :915919.

[80] Mandal M, Breaker RR. Gene regulation by riboswitches. *Nat Rev Mol Cell Biol*, Jun ;5(6) :451-63, 2004.

[81] Ali Nahvi, Narasimhan Sudarsan, Margaret S Ebert, Xiang Zou, Kenneth L Brown and Ronald R Breaker Genetic control by a metabolite binding mRNA. *Chem. Biol.* 9, 10431049, 2002.

[82] Nahvi, A., Barrick, J. E., Breaker, RR. Coenzyme B12 riboswitches are widespread genetic control elements in prokaryotes. *Nucleic Acids Res.* 32, 143150, 2004.

[83] Vitreschak, A. G., Rodionov, D. A., Mironov, A. A., Gelfand, M. S. Regulation of the vitamin B12 metabolism and transport in bacteria by a conserved RNA structural element. *RNA* 9, 10841097 (2003)

[84] Winkler, W., Nahvi, A., Breaker, RR. Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. *Nature* 419, 952956, 2002.

[85] Miranda-Rios, J., Navarro, M., Soberón, M. A conserved RNA structure (thi box) is involved in regulation of thiamin biosynthetic gene expression in bacteria. *Proc. Natl Acad.Sci.* USA 98, 97369741, 2001.

[86] Sudarsan, N., Barrick, J. E., Breaker, RR. Metabolitebinding RNA domains are present in the genes of eukaryotes. *RNA* 9, 644647, 2003.

[87] Rodionov, D. A., Vitreschak, A. G., Mironov, A. A., Gelfand, M. S. Comparative genomics of thiamin biosynthesis in prokaryotes. New genes and regulatory mechanisms. *J. Biol. Chem.* 276, 50935100, 2002.

[88] Mironov, V. N., Perumov, D. A., Kraev, A. S., Stepanov, A. I., Skryabin, K. G. Unusual structure in the regulation region of the Bacillus subtilis riboflavin biosynthesis operon. *Mol. Biol.* 24, 256261, 1990

[89] Mandal, M., Breaker, RR. Adenine riboswitches and geneactivation by disruption of a transcription terminator. *Nature Struct. Mol. Biol.* 11, 2935, 2004.

[90] Grundy, F. J., Henkin, T. M. The S box regulon : a new global transcription termination control system for methionine and cysteine biosynthesis genes in Gram-positive bacteria. *Mol. Microbiol.* 30, 737749, 1998.

[91] McDaniel, B. A. M., Grundy, F. J., Artsimovitch, I., Henkin, T. M. Transcription termination control of the S box system : direct measurement of S-adenosylmethionine by the leader RNA. *Proc. Natl Acad. Sci.* USA 100, 30833088, 2003.

[92] Winkler, W. C., Nahvi, A., Sudarsan, N., Barrick, J. E., Breaker, RR. An mRNA structure that controls gene expression by binding S-adenosylmethionine. *Nature Struct.Biol.* 10, 701707, 2003

[93] Sudarsan, N., Wickiser, J. K., Nakamura, S., Ebert, M. S., Breaker, RR. An mRNA structure in bacteria that controls gene expression by binding lysine. *Genes Dev.* 17, 26882697, 2003.

[94] Sudarsan, N., Wickiser, J. K., Nakamura, S., Ebert, M. S., Breaker, RR. An mRNA structure in bacteria that controls gene expression by binding lysine. *Genes Dev.* 17, 26882697, 2003.

[95] Winkler, W. C., Nahvi, A., Roth, A., Collins, J. A., Breaker, RR. Control of gene expression by a natural metaboliteresponsive ribozyme. *Nature* 428, 281286, 2004.

[96] Jeffrey E. Barrick, Keith A. Corbino, Wade C. Winkler, Ali Nahvi, Maumita Mandal, Jennifer Collins, Mark Lee, Adam Roth, Narasimhan Sudarsan, Inbal Jona, J. Kenneth Wickiser, and Ronald R. Breaker New motifs suggest an expanded scope for riboswitches in bacterial genetic control. *Proc. Natl Acad. Sci.* USA 101, 64216426, 2004.

[97] Voss B, Giegerich R, Rehmsmeier M Complete probabilistic analysis of RNA shapes. *BMC Biol*, 2006.

[98] Abreu-Goodger C, Merino E : RibEx A web server for locating riboswitches and other conserved bacterial regulatory elements. *Nucleic Acids Res*, 33 :W690W692, 2005.

[99] Bengert P, Dandekar T : Riboswitch finder A tool for identification of riboswitch RNAs. *Nucleic Acids Res*, 32 :W154W159, 2004.

[100] Chang TH, Huang HD, Wu LC, Yeh CT, Liu BJ, Horng JT Computational identification of riboswitches based on RNA conserved functional sequences and conformations. *RNA*, 15(7), 2009.

[101] Weinberg Z, Barrick JE, Yao Z, Roth A, Kim JN, Gore J, Wang JX, Lee ER, Block KF, Sudarsan N, Neph S, Tompa M, Ruzzo WL, Breaker RR Identification of 22 candidate structured RNAs in bacteria using the CMfinder comparative genomics pipeline. *Nucleic. Acids. Res.*, 35(14) :48094819, 2007.

[102] Gardner PP, Daub J, Tate JG, Nawrocki EP, Kolbe DL, Lindgreen S, Wilkinson AC, Finn RD, Griffiths-Jones S, Eddy SR, Bateman A Rfam : updates to the RNA families database. *Nucleic. Acids. Res.*, 37(Database) :D136D140, 2009.

[103] C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou. Probcons : Probabilistic consistency-based multiple sequence alignment. *Genome Res.*, 15(2) :330340, February 2005.

[104] R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proceedings of the National Academy of Sciences*, USA, 77(11) :63096313, 1980.

[105] Mathews DH Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA* 10 :1178-1190, 2004.

[106] H. Kiryu, T. Kin, and K. Asai. Robust prediction of consensus secondary structures using averaged base pairing probability matrices. *Bioinformatics*, 23(4) :434441, February 2007.

[107] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29 :11051119, 1990.

[108] Giulio Quarta, Namhee Kim, Joseph A. Izzo and Tamar SchlickJ. Analysis of Riboswitch Structure and Function by an Energy Landscape Framework *J. Mol. Biol.*, 393, 9931003, 2009

[109] M. Zuker, D. H. Mathews and D. H. Turner. Algorithms and Thermodynamics for RNA Secondary Structure Prediction : A Practical Guide In RNA Biochemistry and Biotechnology, 11-43. *NATO ASI Series*, Kluwer Academic Publishers, Dordrecht, NL, 1999

[110] Serganov A, Yuan Y, Pikovskaya O, Polonskaia A, Malinina L, Phan A, Hobartner C, Micura R, Breaker R, Patel D Structural Basis for Discriminative Regulation of Gene Expression by Adenine and Guanine-Sensing mRNAs. *Chem. Biol.* 11(12) :17291741. 2004.

[111] Regulski EE, Breaker RR. In-line probing analysis of riboswitches. *Methods Mol Biol.* 2008 ;419 :53-67.

[112] Gardner PP, Daub J, Tate JG, Nawrocki EP, Kolbe DL, Lindgreen S, Wilkinson AC, Finn RD, Griffiths-Jones S, Eddy SR, Bateman A Rfam : updates to the RNA families database. *Nucleic. Acids. Res.*, 37(Database) :D136D140. 2009.

[113] Peter Clote, Feng Lou, William A. Lorenz Maximum expected accurate structural neighbors of an RNA secondary structure 2011

[114] Lorenz W, Clote P Computing the partition function for kinetically trapped RNA secondary structures. *Public Library of Science One (PLoS ONE)*, 6 :316178, 2011.

[115] Markham NR, Zuker M UNAFold : software for nucleic acid folding and hybridization. *Methods Mol. Biol.*, 453 :331, 2008.

[116] Robert Giegerich, Bjrn Vo, and Marc Rehmsmeier Abstract shapes of RNA *Nucleic Acids Res.*, 32(16) : 48434851, 2004.

[117] Blin G, Denise A, Dulucq S, Herrbach C, Touz H Alignments of RNA structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2010.

[118] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162(3) :705708, December 1982.

[119] L. Holm and C. Sander. Dali : a network tool for protein structure comparison. *Trends Biochem. Sci.*,20(11) :478480,November 1995.

[120] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein. Eng.*, 11(9) :739747, September 1998.

[121] V. A. Ilyin, A. Abyzov, and C. M. Leslin. Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point. *Protein. Sci.*, 13(7) :18651874, July 2004.

[122] J. M. Sauder, J. W. Arthur, and R. L. Dunbrack, Jr. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins.*, 40(1) :622, July 2000.

[123] M. L. Sierk, M. E. Smoot, E. J. Bass, andW. R. Pearson. Improving pairwise sequence alignment accuracy using near-optimal protein sequence alignments. *BMC. Bioinformatics*, 11 :146, 2010.

[124] H.T. Mevissen and M. Vingron. Quantifying the local reliability of a sequence alignment. *Protein Eng.*, 9(2) :127132, 1996.

[125] S. Miyazawa. A reliable sequence alignment method based on probabilities of residue correspondences. *Protein. Eng.*, 8(10) :9991009, October 1995.

[126] Dayhoff, M. O., Schwartz, R. M., Orcutt, B. C. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure* 5 (3) ; 345-352, 1978

[127] Henikoff, S. and Henikoff, J.G Amino Acid Substitution Matrices frome Protein Blocks *PNAS* 89 (22) : 10915-10919, 1992

[128] Reed A Cartwright Logarithmic gap costs decrease alignment accuracy *BMC Bioinformatics* 7 : 527, 2006

[129] M. S. Waterman. Sequence alignments in the neighborhood of the optimum with general application to dynamic programming. *Proc. Natl. Acad. Sci. U.S.A.*, 80(10) :31233124, May 1983.

[130] M. S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J. Mol. Biol.*, 197(4) :723728, October 1987.

[131] M. S. Waterman, M. Eggert, and E. Lander. Parametric sequence comparisons. *Proc. Natl. Acad. Sci.U.S.A.*, 89(13) :60906093, July 1992.

[132] M. Zuker. Suboptimal sequence alignment in molecular biology. Alignment with error analysis. *J. Mol. Biol.*, 221(2) :403420, September 1991.

[133] A. Musacchio, T. Gibson, V. P. Lehto, and M. Saraste. SH3 an abundant protein domain in search of a function. *FEBS Letters*, 307(1) :55 61, 1992.

[134] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403410, October 1990.

[135] U. Muckstein A Variation on Algorithms for Pairwise Global Alignments *Dissertation*, 2001

[136] U. Muckstein, I. L. Hofacker, and P. F. Stadler. Stochastic pairwise alignments. *Bioinformatics*, 18 :S153S160, 2002.

[137] Soper, H.E., Young, A.W., Cave, B.M., Lee, A., Pearson, K. On the distribution of the correlation coefficient in small samples. Appendix II to the papers of "Student" and R. A. Fisher. A cooperative study". *Biometrika*, 11, 328-413. doi :10.1093/biomet/11.4.328, 1917.