Etudes combinatoire et génération aléatoire des structures secondaires d'ARN

Yann Ponty

Etudes combinatoire et génération aléatoire des structures secondaires d'ARN

Mémoire du DEA Algorithmique

Yann Ponty
ponty@lri.fr

Stage dirigé et encadré par Alain Denise
denise@lri.fr

15 Juillet 2003

Laboratoire de Recherche en Informatique

TABLE DES MATIÈRE

ABSTRACT

Ce mémoire résume un travail opéré sous la direction d'Alain Denise de Mars à Juin 2003. Il consiste en une étude des propriétés combinatoires des structures secondaires d'ARN, en rapport avec le problème de la génération aléatoire de structures secondaires d'ARN *réalistes*.

Dans une première partie, nous tenterons de familiariser le lecteur avec le contexte biologique sous-jacent aux problèmes étudiés.

Nous présenterons ensuite un état de l'art de l'étude combinatoire des structures secondaire d'ARN. Pour cela, nous introduirons brièvement une série d'outils théoriques, comme les séries génératrices, et leurs comportements asymptotiques ou les grammaires non contextuelles.

Enfin, nous évoquerons divers mécanismes de génération aléatoire et présenterons une contribution à l'étude des paramètres des structures secondaires d'ARN.

Nous proposerons en annexe une description d'un logiciel dédié à la génération aléatoire de séquences génomiques, GenRGenS, dont l'auteur assure le développement depuis la version 1.0. Nous présenterons aussi un algorithme polynomial de génération aléatoire de chemins culminants, structures combinatoires non algébriques qui apparaissent lors de l'étude d'algorithmes d'alignements de séquence.

1. INTRODUCTION

1.1 Déroulement du stage

La recherche en bioinformatique est l'un des domaines de recherche les plus actifs actuellement. Encore en phase émergeante, les seuls dogmes qu'on y trouve sont ceux issus de la communauté biologique. Tous les algorithmes qui s'y appliquent sont validés par la pratique[1], ce qui laisse de l'espace à des algorithmes heuristiques implémentés avec talent. On y voit donc se multiplier les formalismes, les modèles et les problèmes à une vitesse qui empêche les théoriciens de suivre le rythme. La seule démarche de validation possible pour une équipe ne disposant pas de mathématicien spécialisé dans l'analyse d'algorithme est expérimentale. Ces expériences peuvent avoir lieu *in vivo* ou bien *in silico*.

Mon stage au LRI sous la direction d'Alain Denise est au centre de cette problématique. Il s'agit d'étudier les structures secondaires d'ARN, puis de pratiquer de la génération aléatoire de structures secondaires *réalistes*.

Par réalistes, il faut comprendre *qui plaisent aux biologistes*. Certains d'entre eux ont en effet acquis en un trentaine d'année de fréquentation quotidienne de ces macromolécules une idée *non formalisée* du modèle sous-jacent Ils portent donc en eux une idée beaucoup plus fine de la notion de structure secondaire que l'approche qu'en faisait Waterman en 78[24]. Un des enjeux d'une génération aléatoire de structures secondaires est l'extraction de ce modèle par confrontation à des séquences issues uniformément du modèle. Après des discussions avec Michel Termier (IGM), il est apparu que le modèle privilégie des structures beaucoup trop complexes, c'est à dire que les différentes sous structures d'une structure secondaires sont bien trop courte. Alain Denise a alors décidé d'appliquer un algorithme de génération non uniforme[4] fonctionnant à partir de pondérations. Mon stage consistait donc en une étude des comportements des tailles des sous structures, étude susceptible d'expliquer la complexité des séquences obtenues, et d'en anticiper l'évolution pour des grandes séquences. Ensuite, en une détermination des pondérations à appliquer à l'algorithme[4] pour obtenir des structures plus réalistes.

Dans un premier temps, j'ai cherché à me familiariser avec le contexte biologique. Une étude bibliographique rapide portant sur le problème du repliement de l'ARN a permis une mise en perspective des premières études des structures secondaires d'ARN. Parallèlement, je me suis intéressé à l'avancée des recherches sur la combinatoire des structures secondaires d'ARN. J'ai trouvé en [24, 2, 12, 16] des résultats pour certains types de décompositions. Cependant, ces décompositions me paraissaient insuffisemment expressives, j'ai donc choisi d'en contruire une à partir d'élèments extraits de [24], qui n'avait jusqu'à présent fait l'objet d'aucune étude combinatoire.

J'ai ensuite engendré des structures uniformément de faibles tailles à partir de cette grammaire dans une approche récursive, puis de grandes tailles grâce à la génération de Boltzmann, con-

[1] Et par les talents de communicants de leurs concepteurs …

statant des différences importantes entre les structures engendrées et les structures observées dans la nature. J'ai alors étudié l'asymptotique des différents paramètres structurels, les résultats obtenus confirmant la nécessité d'une génération pondérée. J'ai alors extrait d'une base de données [14] des structures d'ARN ribosomaux que j'ai analysé avec des outils codés pour la circonstance, ce qui m'a permis de définir des objectifs à atteindre dans une génération pondérée.

Afin de me familiariser avec l'adaptation d'un théorème de Drmota[5] centrale à la génération pondérée décrite en[4], j'ai étudié le cas *simple* des structures secondaires, dans lesquelles on souhaite maîtriser le nombre de bases non appariées. Malheureusement, la décomposition trouvée semble induire une grammaire associée ne répondant pas aux hypothèses du théorème de Drmota. On n'a donc pas encore pu produire une pondération garantissant les paramètres statistiques constatés expérimentalement.

Cependant, il semblerait que les conditions d'application du théorème [4] puissent être relâchées. La vérification de cette assertion nécessitent une étude approfondie des propriétés des systèmes d'équations algébriques. C'est l'une des poursuites envisagées à ce stage. Une autre perspective est la transposition du principe de génération de Boltzmann dans l'univers non uniforme. Il s'agirait alors de *déplacer* l'espérance du nombre de lettres à partir de séries multivariées. Enfin, je suis à la recherche de formalismes linguistiques suffisemment puissants pour décrire des pseudonoeuds, et suffisemment restrictifs pour permettre la génération aléatoire. Ces sous structures sont en effet centrales à un grand nombre de phénomènes biologiques, mais ne peuvent être engendrées par une grammaire non contextuelle.

Parallèlement à mon stage, j'ai assisté à Berlin à la conférence RECOMB 2003, où je présentais avec Alain Denise un poster sur la génération aléatoire non uniforme de séquences génomiques, génération implémentée au sein du logiciel GenRGenS, dont j'assure le développement. J'y ai rencontré L.Noe et G.Kucherov, qui m'ont soumis un problème de génération aléatoires de chemin culminants, chemins qui interviennent dans l'étude des biais des algorithmes d'alignements heuristiques. J'ai obtenu un algorithme de complexité cubique pour résoudre ce problème. Il est dans mes projets à cours terme de continuer cette étude sur un plan combinatoire.

1.2 Plan de lecture

Comme l'aura remarqué le lecteur attentif, et non haltérophile, ce mémoire est épais. Cela s'explique peut être par une légère tendance de l'auteur à délayer, mais traduit aussi ma volonté de produire une travail auto suffisant. En effet, apsirant à travailler dans un domaine pluridisciplinaire, j'ai voulu que ce travail soit accessible à un biologiste un peu sensible aux charmes de la combinatoires, et que son but biologique soit rendu clair à un combinatoricien. Il en résulte un pavé peut être un peu indigeste, si l'on veut en lire l'intégralité. Je vais donc décrire rapidement les contenus des différentes parties et les publics auxquels elles sont destinées, afin que chacun puisse sélectionner :

- **Contexte Biologique**
  - **L'ADN :** Rappels de biologie de base
    Public : Mathématicien ou Informaticien
  - **L'ARN :** Rappels de biologie de base
    Public : Mathématicien ou Informaticien
  - **Le Repliement :** Explication et critique de l'algorithme historique
    Public : Tous
- **Combinatoire**
  - **Arsenal :** Rappels de combinatoire énumérative et un peu d'asymptotique
    Public : Informaticien ou Biologiste
  - **Etudes historiques :** Etat de l'art de la combinatoire des ARNs
    Public : Tous
  - **Contribution :** Analyse asymptotique des sous structures dans l'ARN
    Public : Mathématicien ou Informaticien
- **Génération Aléatoire**
  - **Génération et Bioinformatique :** Enjeux de la génération aléatoire en bioinformatique
    Public : Informaticien ou Biologiste
  - **Techniques :** Etat de l'art rapide des techniques de génération aléatoires
    Public : Mathématicien ou Informaticien
  - **Applications :** Application des différentes techniques
    Public : Informaticien ou Biologiste

Partie I

CONTEXTE BIOLOGIQUE

Les données de biologie moléculaire et génomique présentées ici sont le fruit de discussions avec des bio-informaticiens indulgents du LRI (A. Denise, R. Rivière, J.P. Forest, C. Froidevaux, S. Cohen Boulakia ...), et des info-biologistes pédagogues de l'IGM (M. Termier et D. Abergel) que je remercie tous pour leur patience. Une introduction à la biologie moléculaire peut être trouvée dans l'habilitation à diriger des recherches d'Alain Denise [3]. Enfin, je conseille un à lecteur débutant dans le domaine le site de l'institut national de ressources pédagogique[2] (INRP), qui fournit un excellent travail de vulgarisation, et remercie tous les enseignants de fillières PCEM ou SV qui ont pris la peine de mettre *en ligne* leurs cours de biologie moléculaire.

[2] www.inrp.fr

2. L'ACIDE DÉSOXYRIBONUCLÉIQUE (ADN)

L'ADN est le support de l'information génétique. Et, bien que son étude hors-contexte ne suffise pas à expliquer son rôle prédominant dans les mécanismes cellulaires, elle est nécessaire pour envisager un début d'explication. On présentera donc dans un premier temps l'ADN *nu* ou *au repos*, sans se soucier de son interaction avec le milieu. On évoquera ensuite quelques dynamiques dans laquelle il est impliqué, et particulièrement la transcription qui est à l'origine des ARNs.

2.1 Définition bio-moléculaire de l'ADN

L'ADN est un polymère unidimensionnel, c'est à dire une séquence de monomères, les nucléotides.

2.1.1 Les désoxyribonucléotides

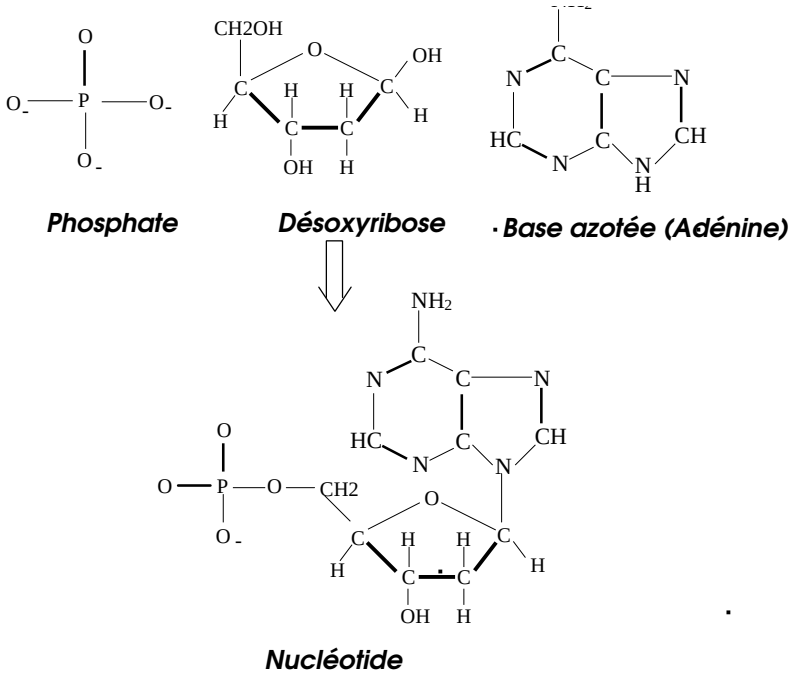


Fig. 2.1 : Constitution d'un désoxyribonucléotide (Adénosine-phosphate)

Un nucléotide est une combinaison covalente d'un acide phosphorique, d'un sucre et d'une base azotée choisie parmi l'Adénine(A), la Guanine(G), la Cytosine(C) et la Thymine(T). Dans le cas de l'ADN, le sucre est un désoxyribose, on parle donc de desoxyribonucléotides . On sépare ces bases en deux catégories, les bases puriques (adénine et guanine), qui contiennent 2 cycles et les base pyrimidiques(cytosine, thymine[1]), qui ne contiennent qu'un cycle.

[1] L'Uracile, substitut dans l'ARN de la thymine, est aussi une base pyrimidique

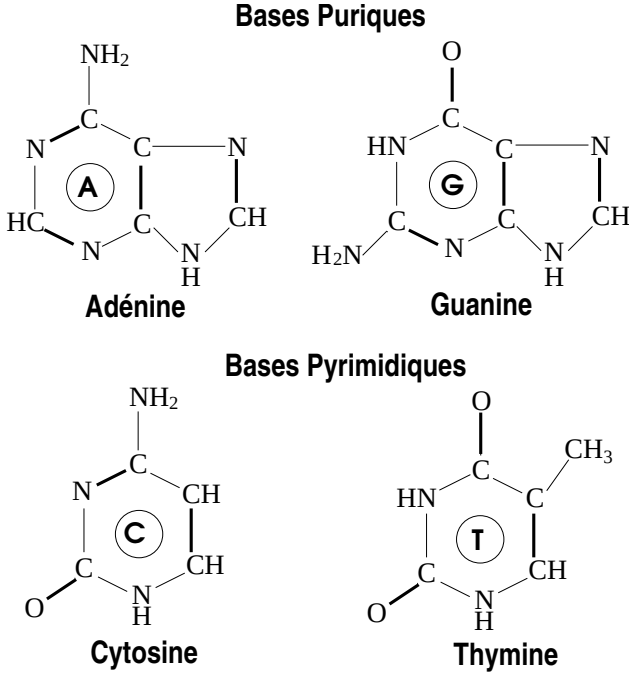


Fig. 2.2 : Les bases azotées

2.1.2 Appariements des bases azotées dans l'ADN

La structure chimique des bases donne lieu à des assemblages de bases dites complémentaires, formant des cycles pyrimidiques/puriques. Les bases complémentaires, classiquement, dans l'ADN sont l'Adénine-Thymine (A/T) et la Cytosine-Guanine (C/G). Les appariements qui en découlent sont appelés appariements Watson Crick. L'union A/T est consacrée par deux liaisons hydrogènes, tandis que l'appariement C/G met en jeu trois liaisons hydrogènes.

2.1.3 La double hélice

Les nucléotides s'empilent dans la séquence au moyen de liaisons phosphodiesters, qui participent à l'assemblage, grace à un phosphate, des désoxyriboses des nucléotides. Les nucléotides s'assemblent ainsi en de longues séquences orientée 5'/3' par héritage des polarités des désoxyriboses, polarités illustrées par la figure 2.4.
La structure tridimensionelle de l'ADN est une double hélice[25] particulièrement stable, régulière et invariante quelle qu'en la composition en bases. Il est donc admis que l'information génétique contenue dans l'ADN est plus à chercher dans la séquence des base que dans sa structure tridimensionnelle ou tertiaire[2]. On étudie donc d'abord la structure primaire de l'ADN, qui correspond à la séquence des bases sur un brin lu dans l'ordre 5' vers 3'. Les bases du brin complémentaires sont entièrement déterminées par complémentation, on peut donc les ignorer lors d'un traitement algorithmique. En effet, dans l'ADN, les seuls appariements possibles sont A/T et C/G pour des problèmes de stabilité et d'encombrement stérique.

[2] Cependant, l'étude de l'enroulement de la double hélice autours d'un histone, appelé nucléosome, à permis de mettre en évidence des interactions entre paires de bases proches géographiquement, ce qui plaide pour une importance de la structure tertiaire de l'ADN. Notemment, des périodicités de 200bp sont observées dans les séquences, qui qui correspond à la circonférence des nucléosomes.

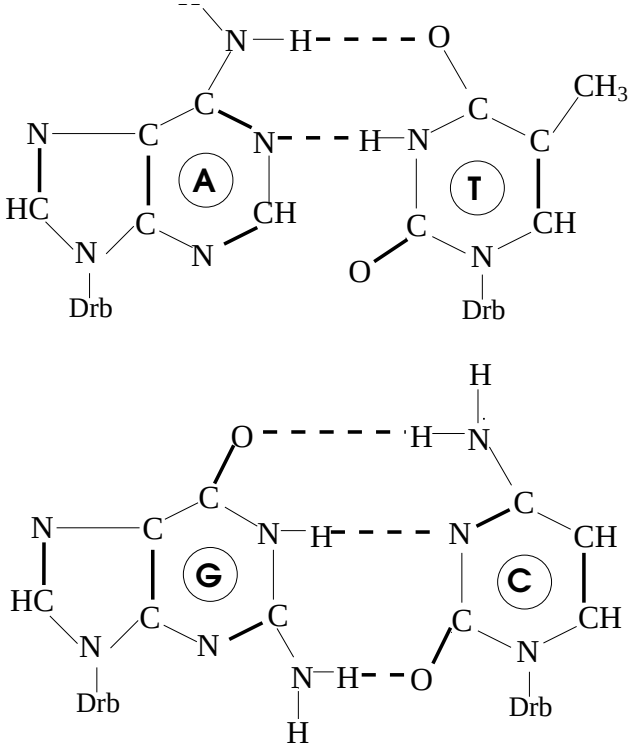


Fig. 2.3 : Les appariements Watson Crick.

2.2 La réplication

Lors de la mitose, les deux brins de l'ADN sont séparés par une enzyme, l'hélicase (voire figure 2.5, 1). Son travail est complété par des protéines de liaisons (2), qui empèchent les deux brins de se recoller derrière l'hélicase. Les complémentaires, aussi appelés molécules filles, des deux brins ne peuvent, pour des raisons chimiques, être générés que dans le sens 5'→3'. Le brin père 3' est donc complété séquentiellement par la DNA-polymèrase(3). Le brin 5' nécessite quant à lui la création, toujours par la DNA-polymérase, de petites séquences, les fragments d'Okazaki, qui seront ensuite recollés par une ligase(4).
La replication est, toujours chez les procaryotes et très souvent chez les eucaryotes, bidirectionelle et initiée à partir de sites d'amorçage, des séquences de bases spécifiques. L'intervention d'une enzyme, la télomèrase, est ensuite nécessaire pour reconstituer les extremités, appelées télomères, altérées par le mécanisme de réplication. Les télomères sont donc reconstitués de façon quasi systèmatique par des copies d'un fragment d'ARN puis se replient en boucle pour éviter toute dégradation de l'ADN.

2.3 La transcription

On a jusqu'ici évoqué le support et la copie de l'information génétique, il convient maintenant d'en expliquer l'expression. L'ADN s'exprime par l'intermédiaire des ARNs, auxquels nous consacreront le prochain chapitre, molécules qui sont créées au cours de la phase de transcription. Seules certaines portions de l'ADN sont transcrites, ces séquences sont appelés gènes.
Lors de la phase de transcription, un complexe protéique,l'ARN polymérase, se fixe sur l'ADN. On appelle promoteur le site de fixation de l'ARN polymérase. Il est de taille variable, et peut associer plusieurs séquences cibles. La plus fréquente est la séquence TATAAT, qui est propice à

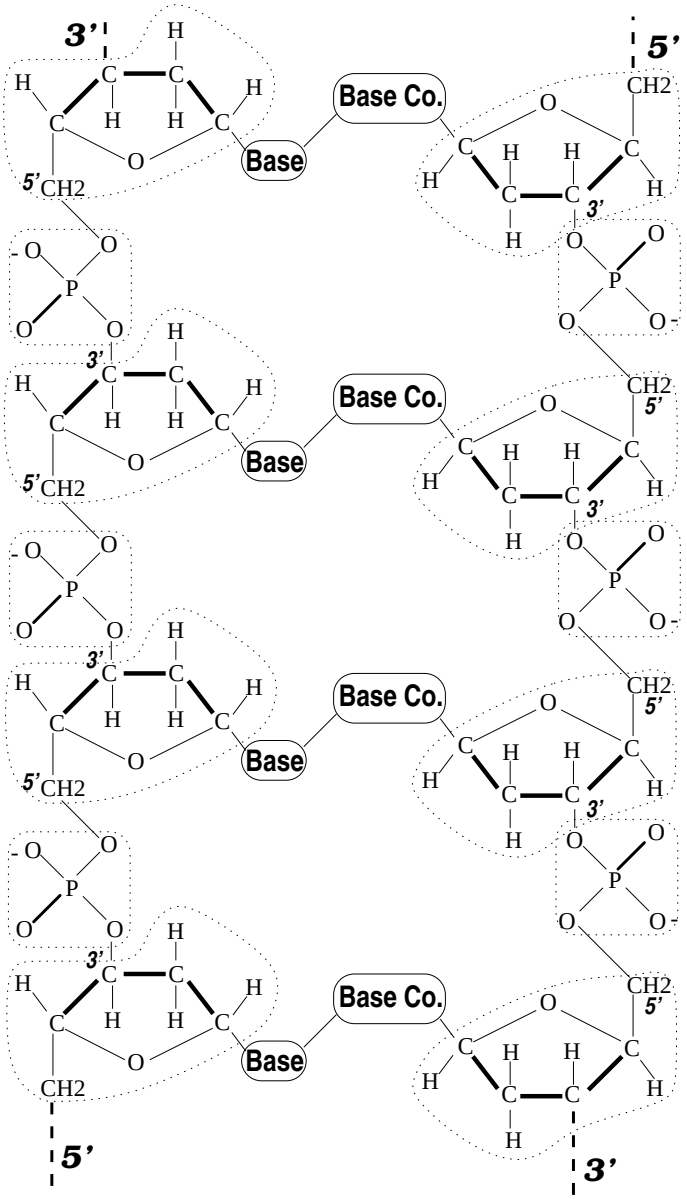


Fig. 2.4 : Assemblages de nucléotides formant la double hélice.

l'écartement des deux brins.La fixation de l'ARN polymèrase sur l'ADN peut être activée ou inhibée par la présence de protéines, appelées facteurs régulateurs de transcription.
Une fois l'ARN polymèrase fixée, elle réalise une copie complémentaire d'une portion du brin sens de l"ADN dans le sens 5'→3', en complémentant le brin anti-sens dans le sens 3'→5' (voire Figure 2.7). Elle substitue au passage l'uracile à la thymine. La transcription est stopée quand l'ARN polymérase rencontre un site particulier, appelé terminateur, ou sur intervention de facteurs protéiques chez cerains procaryotes[3]. Le polymère obtenu est appelé transcrit primaire de l'ARN ou préARN. Ce transcrit présente une segmentation en 3 régions : 5'UTR, ORF, 3'UTR, l'ORF pouvant être interrompue par des introns
Chez les eucaryotes, il existe quelques aménagements au cadre général de la transcription :

[3] Facteur rho chez E. Coli.

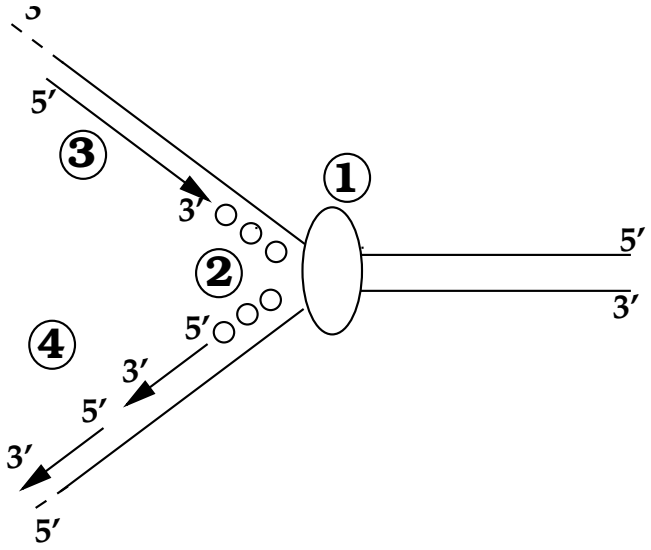


Fig. 2.5 : Replication de l'ADN.

- Chaque classe d'ARNs est engendrée par une ARN polymérase spécifique.

- Avant d'être fonctionnel hors du noyau, le transcrit primaire doit faire l'objet d'une maturation. La maturation consiste, classiquement, en l'adjonction d'une *coiffe*[4] en 5', d'une queue polyadénilée en 3'[5], ainsi qu'en un épissage du transcrit primaire. Celui est en ef-

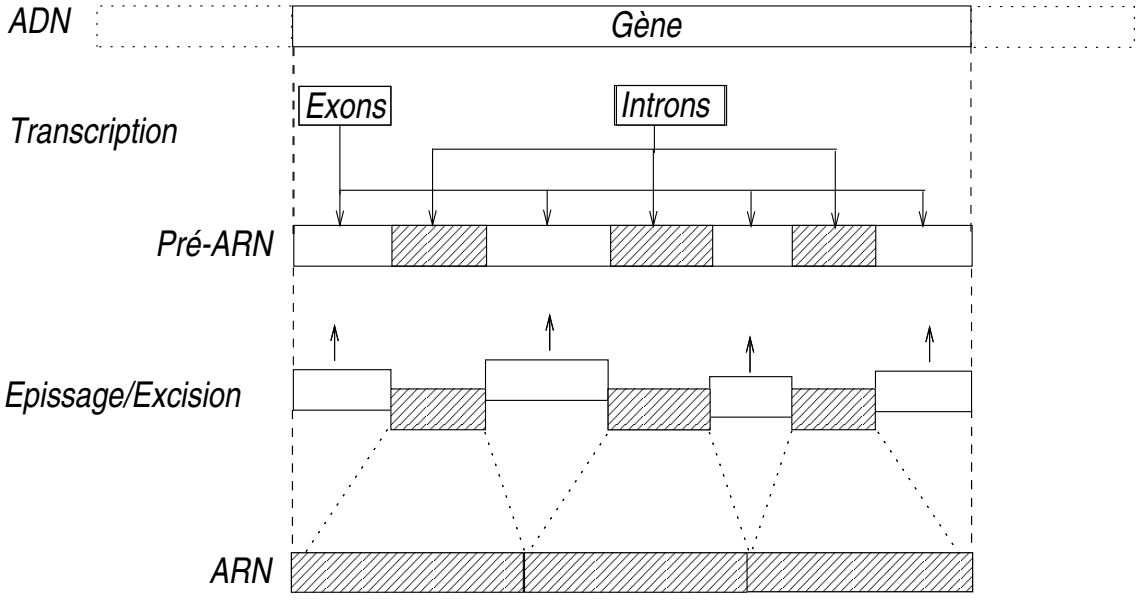


Fig. 2.6 : L'épissage du transcrit primaire chez les eucaryotes

fet constitué d'une alternance d'exons, matériel génétique codant, et d'introns non codants. L'épissage est le mécanisme par lequel les introns sont éliminés du transcrit pri-

[4] Une coiffe est la somme d'un groupement triphosphate et d'une base purique (A ou G).
[5] Sauf chez les ARNs codant pour des histones, protéines à la base du nucléosome.

maire comme le montre la figure 2.6 et les introns sont concaténés.

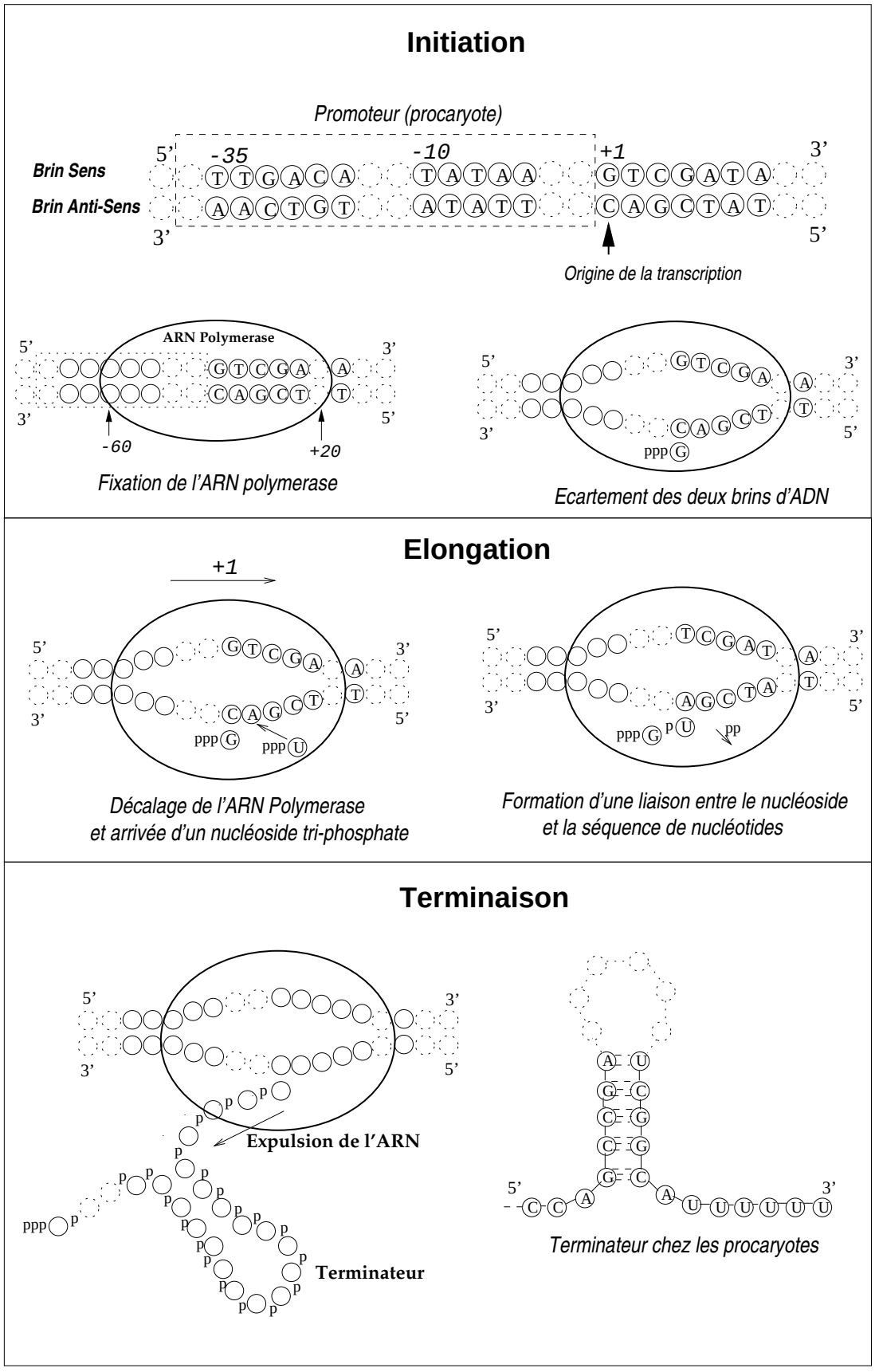


Fig. 2.7 : Les étapes de la transcription

3. L'ACIDE RIBONUCLÉIQUE (ARN)

L'ADN est circonscrit au noyau de la cellule chez les eucaryotes. Or l'expression phénotypique du matériel génétique passe essentiellement par l'action de protéines, qui sont localisées hors du noyau, dans le cytoplasme.
L'ARN est donc l'intermédiaire par lequel l'ADN s'exprime en dehors du noyau. Il est synthétisé à partir d'un segment d'ADN, le gène.
Tout le contenu informatif de l'ADN est transcrit en un ARN qui sera soit un acteur (ARNs ribosomaux ou ARN de transfert), soit une information *brute* nécessitant étape supplémentaire de traduction pour s'exprimer (ARN messagers).

3.1 Définition bio-moléculaire de l'ARN

La molécule d'ARN est, comme l'ADN, un polymère linéaire dont les monomères sont cette fois ci des ribonucléotides, par substitution du sucre β-D-ribose au désoxyribose. Comme le montre la

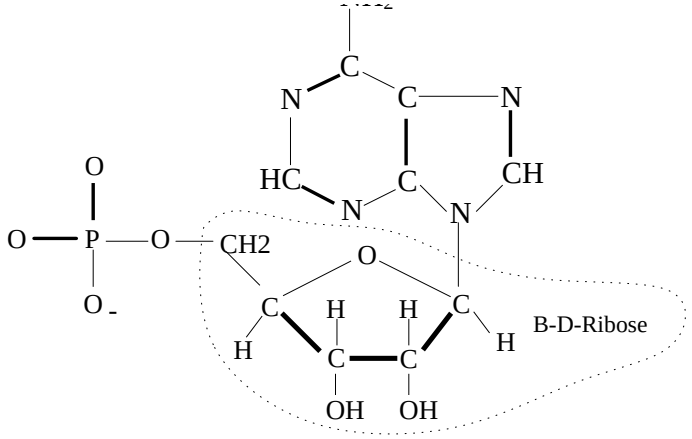


Fig. 3.1 : Un ribonucléotide

figure 3.1, un ribonucléotide estt la combinaison covalente d'un phosphate, d'un ribose et d'une base azotée. Les bases azotées disponibles pour l'ARN sont l'Uracile, l'Adénine, la Guanine et la Cytosine. L'orientation de la molécule d'ARN est la même que celle de l'ADN, du 5' au 3'.

3.2 Origine de la structure secondaire des ARNs

3.2.1 Définition des différentes structures d'une molécule

On caractérise l'organisation dans l'espace des macromolécules biologiques (ADN, ARN ou protéines) par leurs structures primaire, secondaire et tertiaire.

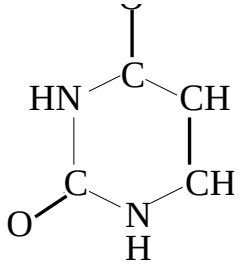


Fig. 3.2 : L'uracile, substitut de la thymine dans les ARNs

Structure primaire

Fig. 3.3 : Structure primaire d'ARNr 16S de Pyrococcus Furiosus

La structure primaire est la séquence des bases lues dans le sens 5'→3'.

Structure secondaire

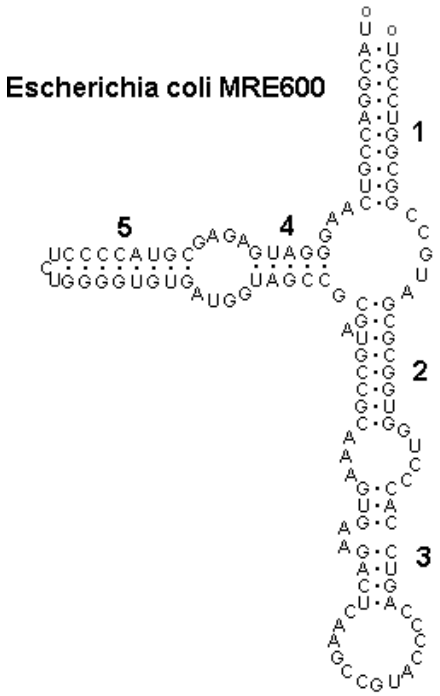


Fig. 3.4 : Structure secondaire de l'ARNr 5s d'E. Coli

Il n'existe pas de consensus total sur la définition des structures secondaires.
On peut cependant caractériser la structure secondaire d'un ARN par un ensemble de liaisons

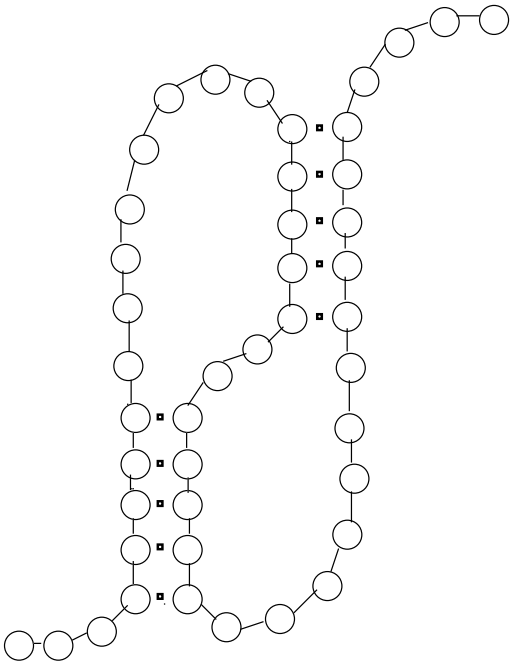


Fig. 3.5 : Un pseudo noeud

entre ses bases. De plus, afin de respecter une correspondance entre type de structure et dimension de l'espace requis pour les dessiner[1], on considère qu'une structure secondaire est planaire. Au delà de ce plus petit dénominateur commun, il reste à définir les types de liaisons autorisées, ainsi que la tolérance ou non de pseudo-noeuds (voir Figure 3.5). Il circule dans la communauté bioinformaticienne autant de définitions des structures secondaires que de variations planaires [2] sur ces deux thèmes. Nous imposons donc la définition suivante, inspirée de [24].

Définition 1 (Structure secondaire de l'ARN) :
On appelle structure secondaire d'un ARN la structure planaire comportant les appariements Watson-Crick et Wobble, et ne comportant pas de pseudo-noeuds.

Structure tertiaire

La structure tertiaire, est la localisation des constituants chimiques de la molécule dans l'espace. Là encore, certains se cantonne à la topologie de la molécule quand d'autres lui préfèrent les positions relatives des bases dans l'espace tridimensionel.

Structure quaternaire

On évoque parfois la structure quaternaire d'une molécule en indiquant son positionnement relativement à des molécules avec lesquelles elle est liée physiquement pour permettre une fonction.

[1] primaire ⇔ linéaire (1D)
secondaire ⇔ planaire (2D)
tertiaire ⇔ tridimensionnelle (3D)

[2] Des pseudos noeuds *imbriqués* ne sont plus planaires. Certaines définitions admettent donc la présence de pseudo-noeuds *raisonnable*, c'est à dire planaire.

Fig. 3.6 : Structure tertiaire d'un ARNt

Fig. 3.7 : Structure quaternaire des histones

C'est le cas de la structure tridimensionnelle du nucléosome (voir Figure3.7), décrite comme la

structure quaternaire des histones [3].

3.2.2 Repliement de l'ARN

Les structures secondaires et tertiaires de l'ARN sont bien plus variées et porteuses d'informations fonctionelles que celles de l'ADN. En effet, la structure simple brin de l'ARN permet un repliement

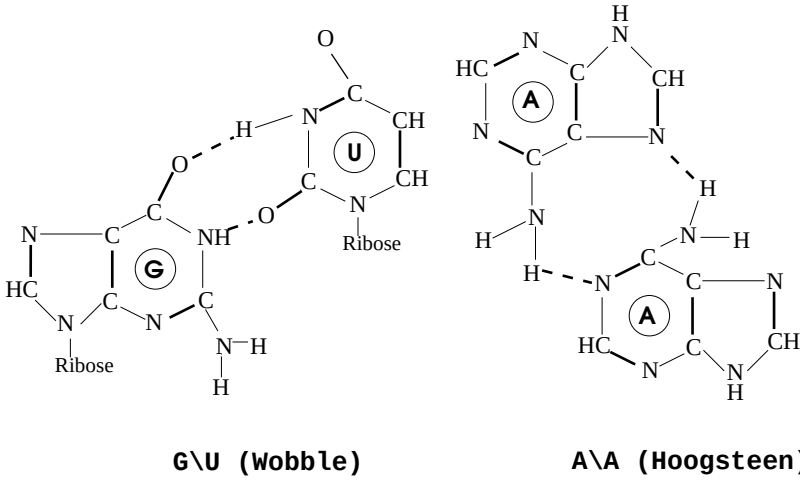


Fig. 3.8 : Quelques appariements non Watson Crick observables chez certains ARNs

de la molécule sur elle même, au moyen d'appariements Watson-Crick(voir 2.3 en substituant l'uracile à la thymine), mais aussi d'appariements Wobble ou Hoogsteen (voir 3.8). On trouvera en [15] un travail de classification des appariements[4].
Le système de liaisons ainsi défini, associé aux dispositions des ribonucléotides dans les différents types d'appariements, définit la conformation spatiale de l'ARN.
Dans de nombreux cas, on met en évidence une relation entre cette structure de l'ARN et son intervention dans les mécanismes biologiques. Cette structure est donc porteuse d'information sur la fonction de l'ARN, au même titre que la séquence des bases.
De plus, il n'existe pas d'inclusion stricte entre les contenus informatifs de la séquence et ceux de la structure. En effet, on ne peut pas associer à une conformation spatiale donnée une unique structure primaire, de même qu'il semble impossible d'établir le repliement tridimensionnel de l'ARN avec certitude à partir de la structure primaire(voir 4.1.2, page 30). L'étude conjointe de la séquence de bases et de la structure semble donc nécessaire à la compréhension du rôle d'un ARN. La complexité, prise ici au sens algorithmique, de l'étude de la structure tertiaire étant rédhibitoire[5], on ne considère souvent que la structure secondaire d'un ARN quand on veut prendre en compte sa topologie.

[3] La structure tridimensionelle du nucléosome est parfois décrite comme la structure quaternaire des histones.

[4] Propre à dégouter le plus témeraire des bioinformaciens. Presque tous les appariements et triplets semblent admissibles. On est donc bien loin du confort tout relatif des séquences primaires d'hélices palindromiques ... Un lecteur informaticien ayant le coeur solidement accroché pourra aussi consulter à l'adresse `http://prion.bchs.uh.edu/bp_type/bp_structure.html` une encyclopédie des appariements et triplets.

[5] L'apparition d'interaction tertiaire simple, comme les pseudo-noeuds fait sortir les structures étudiées de l'univers *hors contexte*, ce qui pénalise un traitement récursif. Par exemple, le repliement de l'ARN avec des pseudo noeuds généralisés est NP-Complet

3.3 Les ARNs : fonctions et structures

On distingue les différents types d'ARNs selon les rôles qu'ils jouent au sein de la cellule.

3.3.1 Les ARNS messagers (ARNm)

Ils représentent moins de 5% des ARNs cellulaires.
Rôle : Ils amènent l'information extraite de l'ADN au ribosome, qui va exprimer cette information par la synthèse protéique.
Structure secondaire : Les structures secondaires des ARNs messagers semblent assez variées, aucun modèle fortement contraignant n'a jusqu'ici été formulé.
Jusqu'à présent, on considérait que la structure des ARNm était négligeable. En effet, au cours de la traduction, l'ARN est étiré en une structure linéaire, ce qui semblait plaider pour une nullité du rôle joué par sa structure. Cependant, des études récentes sur le phénomère d'interférence d'ARN confère à l'ARNm un rôle qui dépasse celui d'intermédiaire et qui ferait intervenir sa structure. Cet intérêt pour la structure de l'ARNm étant assez récent, il n'existe pour l'instant que relativement peu de structures secondaires connues[6].

3.3.2 Les ARNS de transfert (ARNt)

Ils représentent environ 15% des ARNs cellulaires. Leur structure primaire est très variable.
Rôle : Les ARNt sont des molécules qui traduisent les codons des ARNm en acides aminés. De structures fortement contraintes et quasi identiques, ils sont caractérisés par leurs anticodons(voir Figure 3.9). Un enzyme spécifique à chaque anticodon, l'aminoacyl-ARNt synthétase, se charge d'adjoindre à chaque ARNt l'acide aminé correspondant au complémentaire de l'anticodon. Cet acide aminé est collé sur le brin 3' de l'ARNt au niveau d'un triplet CCA. Lors de la traduction, l'anticodon va s'apparier avec le codon de l'ARNm et permettre la concaténation d'un acide aminé adéquat.
Structure secondaire : Comme l'illustre la Figure 3.9, la structure secondaire de l'ARNt est très fortement contrainte. Sa forme de trèfle est invariante, et seule la dimension du renflement(Boucle V de la Figure3.9) permet de distinguer les structures secondaires de différents ARNt.
Remarque : On rencontre dans les ARNt des bases inhabituelles chez les ARNs (pseudouridine, inosine ou encore des bases méthylées) qui sont en fait des bases modifiées après la transcription afin de conserver la structure tridimensionnelle de l'ARNt.

3.3.3 Les ARNS ribosomiques (ARNr)

Ils représentent près de 80% des ARNs cellulaires.
Rôle : Ils forment, avec des protéines spécialisées, le ribosome, qui est le siége de la traduction. Ils sont caractérisés par leur constante de sédimentation, exprimée selon une unité S(de Svedberg).
Structure secondaire : Leurs structures secondaires sont assez variées, même parmis des d'ARNr ayant une même constante de sédimentation.

3.4 La traduction

La traduction est le mécanisme par lequel une protéine est engendrée à partir des informations contenues dans un ARN messager. Elle met en jeu une macro molécule complexe, le ribosome. Dans la suite, on appellera codon un triplet de nucléotides.

[6] La fiabilité des structures prédites *in silico* étant, pour l'instant, sujette à caution ...

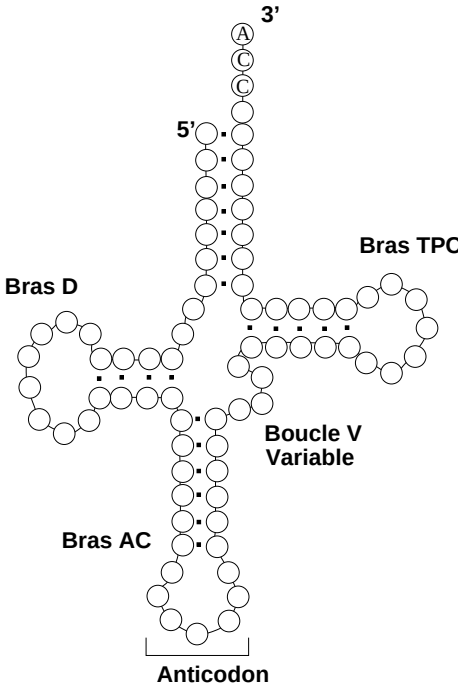


Fig. 3.9 : Structure secondaire d'un ARNt de levure

Famille	*Localisation*	**Nom**	Taille
Procaryotes($70S$)	*Grande Sous-Unité*($50S$)	**5S**	120 nt
		23S	2900 nt
	Petite Sous-Unité($30S$)	**16S**	1542 nt
Eucaryotes($80S$)	*Grande Sous-Unité*($60S$)	**5S**	120 nt
		5,8S	160 nt
		28S	4800 nt
	Petite Sous-Unité($40S$)	**18S**	1900 nt

Fig. 3.10 : Les différents ARNr

3.4.1 Le ribosome

Le ribosome est constitué de matériel mixte ARNr/protéines. On le décompose en deux partie, appelées petite et grande sous unités. Chacune des sous unités est composée d'ARNm(s) et de protéines. Hors de la phase de traduction, les sous unités des ribosomes sont séparés par la présence

Famille	*Sous-Unité*	**ARNr**	Nombre de protéines
Procaryotes	*Grande Sous-Unité*	**5S+23S**	34
	Petite Sous-Unité	**16S**	21
Eucaryotes	*Grande Sous-Unité*	**5S+5,8S+28S**	49
	Petite Sous-Unité	**18S**	33

Fig. 3.11 : Les compositions des ribosomes

de protéines.

3.4.2 Initiation

Grossiérement, on peut dire que la petite sous unité du ribosome se fixe sur l'ARNm, ce qui permet à la plus grande sous unité de se coller à elle en entourant l'ARNm.
Le début de la traduction d'un ARN messager se fait à partir d'un codon spécifique, ou codon START(AUG,GUG ou UUG).
Chez les eucaryotes, le ribosome se fixe en amont du codon START. Il glisse alors dans le sens 5'→3' en scannant l'ARNm à la recherche d'un codon START.
Chez les procaryotes, l'ARN 16S se fixe sur l'ARNm en amont de moins de 10 bases du codon START. Cette zone contient un motif dit de Shine-Dalgarno qui correspond au complémentaire du 16S.
Une fois le codon START trouvé, un ARNt chargé spécifique de l'initation, l'ARNti, se fixe sur le codon START et initie la traduction.

3.4.3 Elongation

Pendant la phase d'élongation, les acides aminés sont concaténés pour former des protéines. L'élongation est une séquence de formations de liaison peptidiques et de translocations.
Tout d'abord, le ribosome permet la fixation d'un ARNt chargé dans le site A. Une liaison peptidique se forme alors entre les acides aminées fixés à l'ARNt présents sur le site P et celui fixé sur l'ARNt de A.
Intervient alors une étape de translocation, au cours de laquelle le complexe formé par l'ARNt de A et sa chaîne peptidique sont décallés sur le site P par une translation du ribosome d'exactement un triplet. L'ARNt initialement présent en A hérite alors de la chaîne peptidique présente en P. L'ARNt déchargé est alors éjecté. Le site A, libre et exposé à un nouveau codon, provoque l'arrivée d'un nouvel ARNt porteur de son anti codon et de son acide aminé correspondant. On se retrouve alors dans les conditions initiales de la translocation, qui se reproduit donc jusqu'à la terminaison.

3.4.4 Terminaison

La terminaison se produit quand le ribosome rencontre un codon STOP (UAG, UAA et UGA), qui ne correspondent à aucun anticodon d'ARNt. Le site A, vide, est disponible pour un facteur protéique RF (Release Factor) qui permet une dernière translocation. Cette translocation provoque une coupure entre l'ARNt présent en P et la chaîne polypeptidique dont il est le porteur. Les sous unités du ribosomes se séparent alors.

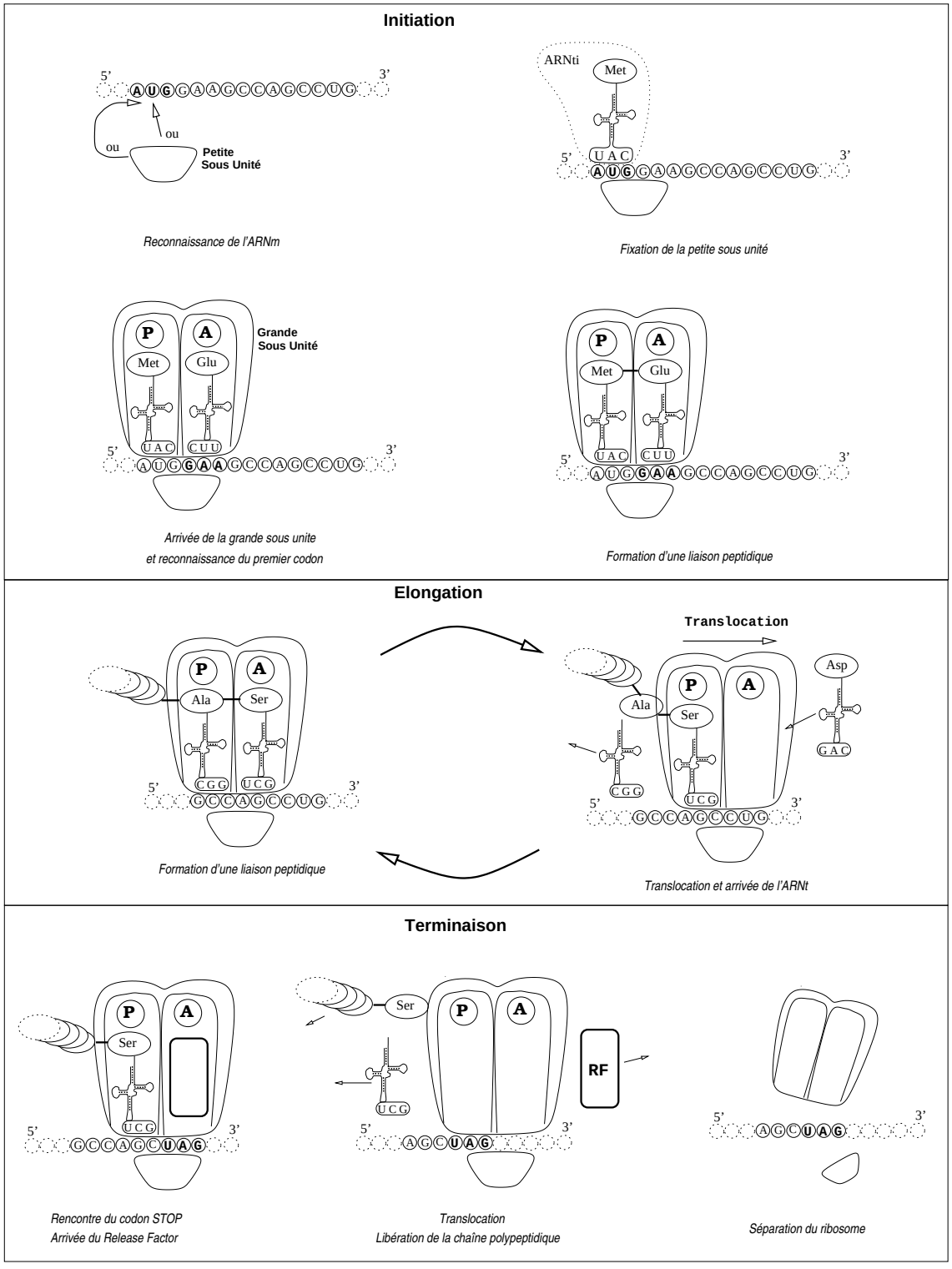


Fig. 3.12 : Les phases de la traduction

4. LE REPLIEMENT DES ARNS : UN EXEMPLE DE PROBLÈME BIOINFORMATIQUE ...

Il s'agit, à partir de la séquence des bases d'un ARN lues dans le sens $5' \rightarrow 3'$, d'inférer sa structure secondaire la plus probable dans un modèle donné.C'est ce problème qui est à l'origine de l'article de Waterman en 78, qui jette les bases d'une étude combinatoire des ARNs. On propose donc dans cette partie un petit tour d'horizon des techniques de repliement.

4.1 Approche thermodynamique

Le milieu cellulaire est dense, l'ARN cotoie donc un grand nombre d'acteurs de la machinerie cellulaire. Parmis ceux ci, d'autres ARNs présentant des bases non appariées, auxquels il peut être tenté de se concaténer, altérant ainsi son contenu ou inhibant sa fonction. L'ARN se doit donc d'être compact afin de survivre et de s'exprimer. En particulier, il faut que son energie libre, notion de thermodynamique à mettre en relation avec le nombre de bases non appariées, soit la plus faible possible. C'est cette quantité que l'approche thermodynamique du repliement des structures secondaires souhaite minimiser. La solution de M. Zucker[28], implémentée par le logiciel MFold, consiste à employer un algorithme de programmation dynamique.

4.1.1 Présentation

Cette technique de repliement est rendue possible par une décomposition des structures secondaires, qui permet la mise en place d'un algorithme de programmation dynamique, la décomposition en K-Boucles.
On suppose les bases de la structure secondaire d'un ARN numérotées de 1 à n dans le sens $5' \rightarrow 3'$.
On dit d'une base l qu'elle est accessible depuis une paire de base (i, j) appariées si $i < l < j$.

Définition 2 (K-Boucles) :
La boucle B d'une paire (i, j) de bases appariées est l'ensemble des bases l telles que :

- *l est accessible depuis (i, j).*
- *$\forall (m, n)$ appariées accessibles depuis (i, j), l n'est pas accessible depuis (m, n).*

On appelle $l_s(B)$(resp. $l_d(B)$) le nombre de bases non appariées (resp. le nombre de paires de bases) de B. B est une K-Boucle si elle contient $K - 1 (= l_d(B))$ paires de boucles.

Intuitivement, la K-Boucle d'une paire b de base est l'ensemble des bases accessibles à partir de b en parcourant la structure secondaire dans le sens de l'arborescence sans traverser de paire de bases appariées.

Proposition 1 (Décomposition en K-Boucles) :
Chaque structure secondaire S de taille n est décomposable de façon unique en K-boucles $B_0, B_1, \ldots, B_m$ où m est le nombre de paires de bases de appariées dans S
et B_0 la boucle relative à la paire virtuelle $(0, n + 1)$, aussi appelée boucle extérieure.

A partir de cette décomposition, Zucker propose un modèle additif pour l'énergie libre E d'une structure secondaire d'ARN.

$$E = \sum_{i=0}^{m} e(B_i)$$

où $e(B_i)$ est l'energie libre d'une boucle.
Ce modèle est le modèle du *plus proche voisin*, car les boucles relatives à une paire de base ne contiennent que des paires de bases appariées voisines dans la structure secondaire.
Il reste encore à définir l'énergie libre d'une K-boucle, celle ci dépendra de son paramètre K :

- **K=1 : Les Tiges Boucles**
  Aucune paire de bases appariées n'apparait dans la 1-boucle de la paire (i, j). On appelle

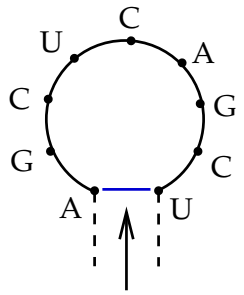


Fig. 4.1 : Une tige boucle

ces structures des tiges-boucles. Des résultats de théorie des polymères établissent que l'energie libre $\delta\delta G$ d'une tige boucle est de la forme :

$$\delta\delta G = 1.75 * R * T * ln(l_s(B))$$

D'autres facteurs viennent compléter cette energie libre *de base*. Par exemple, un bonus est attribué à la tige boucle si les deux premières bases non appariées ont des affinités. D'autre part, les energies libres des tiges-boucles telles que $l_s(B) = 3$ ou 4 sont ajustées à des mesures expeérimentales.

- **K=2 :**

Les paires de bases
Lorsque qu'une paire (i, j) est suivie de deux bases appariées selon un appariement

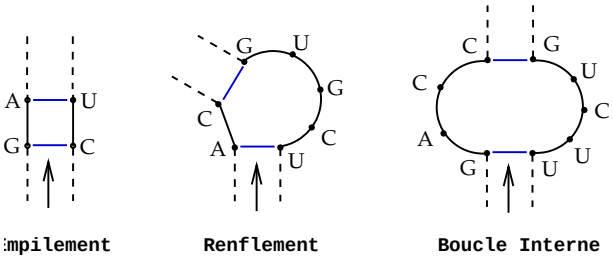


Fig. 4.2 : Représentation des 2-boucles

autorisé(A/U, C/G et G/U), les energies de telles structures sont extraites d'une base de données experimentale, dépendant de la température.

Les renflements
Un renflement est une séquence de bases non appariées séparant sur l'un des brins deux paires de bases appariées successives dans la structures. L'energie libre d'une telle sous structure dépend-uniquement du nombre de bases non appariées de la 2-Boucle et de la température. Ici encore, les énergies libres sont puisées dans un stock de données expérimentales et completées par interpolation selon une formule proche de celle des tiges boucles.

Les boucles internes
Une boucle interne est un couple de paires de bases séparées par des bases non appariées sur les deux brins.
Ici, l'energie libre dépend à la fois du nombre de bases non appariées, mais aussi de la dissimétrie de la structure. Enfin, des bonus et malus sont attribués selon la composition des premiers couples de bases non appariées dans les deux sens de lecture.

- **K>2 : Les multiboucles**
  Les multiboucles sont les noeuds de l'arborescence de la structure secondaire. La K-boucle

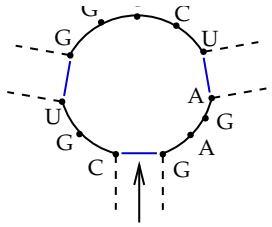


Fig. 4.3 : Les multiboucles

correspondant contient toutes les bases non appariées, ainsi que les premières paires de bases de chacune des empilements de paires de bases, de la multiboucle. En l'absence de modèle thermodynamique spécifique à ce type de structure, M.Zucker et al. choisissent une formule pour l'energie libre qui simplifie les calculs :

$$\delta\delta G = a + b * l_s(B) + c * l_d(B) + \delta\delta G_{stack}$$

où $\delta\delta G_{stack}$ est la somme des incréments d'energie libre dus aux interactions entre les début des empilements de paires de bases et leurs plus proches voisins non appariés, constatés expérimentalement.

Soit $W(i,j)$ l'energie minimale d'un repliement de la séquence $[i,j]$,
$V(i,j)$ l'energie minimale de la séquence $[i,j]$ contenant la paire (i,j),
$e_t(i,j)$(resp. $e_e(i,j)$) l'energie libre d'une tige-boucle(resp. d'un empilement de deux paires de bases) commençant en (i,j)
et $e_{ri}(i,j,i',j')$ l'energie libre d'un renflement ou d'une boucle interne délimitée par les paires (i,j) et (i',j'), avec i' et j' accessibles à partir de (i,j).
On déduit alors du modèle thermodynamique ci dessus les équations suivantes :

$$W(i,j) = \begin{cases} \infty & Si \quad j-i<4 \\ Min(W(i+1,j), W(i,j-1), V(i,j), & \\ Min_{i\leq k<j}(W(i,k)+W(k+1,j))) & Sinon \end{cases}$$

$$V(i,j) = \begin{cases} \infty & Si \quad j-i<4 \\ Min(e_t(i,j), e_e(i,j) + V(i+1,j-1) & \\ VBI(i,j), VM(i,j)) & Sinon \end{cases}$$

où

$$VBI(i,j) = Min_{i<i'<j'<ji'-i+j-j'>2}\{e_{ri}(i,j,i',j') + V(i',j')\})$$

et

$$VM(i,j) = Min_{i<k<j-1}\{W(i+1,k) + W(k+1,j-1)\} + a)$$

qui peuvent être résolues par programmation dynamique en temps $O(n^3)$ en limitant la taille des structures de VBI à une taille *réaliste* de 30 bases non appariées.
Ces équations constituent une mise en oeuvre simplifiée du modèle ci dessus. En effet, elles supposent que toutes les boucles multiples ayant le même degrès K auront une même energie libre. De plus, elles interdissent les empilements de bases de taille 1. On se refèrera aux différents articles de Zucker à ce sujet pour obtenir des versions plus complètes (et assez indigestes) de ces équations.

4.1.2 Critique

Les travaux de Zucker sur l'inférence de la structure secondaire d'ARN par minimisation de l'énergie libre mettent en évidence l'insuffisance d'une approche strictement physico-chimique. En effet, les structures secondaires constatées expérimentalement sont souvent parmis les plus stables, mais sont rarement la structure renvoyée par les premières versions de MFold. Trois hypothèses non mutuellement exclusives sont susceptibles d'expliquer ce phénomène :

- Le modèle d'interaction *plus proche voisin*, en négligeant au moins une dimension du problème ainsi qu'en limitant les interactions à des complexes à priori proche, est insuffisant.
- L'ARN étant transcrit séquentiellement, il se forme des structures stables minimisant l'énergie libre au sein du *préfixe* déjà généré. L'émission d'une nouvelle base ne remettra en question la structure préfixe que si celle ci est optimisable au delà d'un seuil.
- L'ARN n'étant pas isolé, sa proximité avec des ARNs et autres agents de la molécule induit un repliement optimal dans son contexte, mais pas au regard d'une minimisation de l'energie libre.

Dans des travaux ultérieurs, Zucker s'est donc attelé à renvoyer un ensemble de structures sous optimales, parmis lesquelles les structures observées experimentalement apparaissent parfois.

4.2 Autres approches

Des travaux de J. Waldispühl *et al* [23] ont ramené la recherche des sous optimaux à un problème d'analyse syntaxique basée sur une grammaire S-attribuée ambiguë. On peut aussi trouver en [11] les bases du repliement, optimal cette fois, vu comme un problème de parsing.

Une approche alternative, qualifiée de *comparative* est utilisée par S. Engelen, F. Tahi et M. Régnier dans [21]. On délimite les parties palindromiques en observant des corrélations entre les mutations à certaines positions. Partant du principe que la structure a un impact sur la fonction des ARNs, on interprète ces corrélations de la façon suivante : Si deux bases sont appariées, et que leur appariement est important fonctionnellement, alors la mutation d'une base provoque

la mutation de la base appariée pour que l'appariement soit conservé. A partir d'un grand nombre de séquences homologues, on peut prédire les palindromes dans l'ARN, et donc en deviner la structure secondaire.

A chaque nouvelle approche du problème de repliement apparaîssent de nouvelles raisons d'étudier les propriétés combinatoires des structures d'ARN, ce que nous nous proposons d'étudier dans la prochaine partie.

Partie II

ETUDES COMBINATOIRES DES STRUCTURES SECONDAIRES D'ARN

5. ARSENAL COMBINATOIRE

Dans ce chapitre, on va introduire l'outil principal de la combinatoire, les séries génératrices. On donnera ensuite quelques techniques pour extraire l'information qu'elles contiennent. On finira par une présentation du comportement asymptotique des coefficients de deux grandes familles de séries génératrices :

- *Les séries rationelles*, qui sont les séries génératrices des langages rationnels.
- *Les séries algébriques*, qui sont les séries génératrices des langages non contextuels.

L'ensemble des langages rationels est inlcus dans l'ensemble des langages algébriques, ce qui peut sembler diminuer l'intérêt d'une étude séparée de ces deux classes. Cependant, on obtient sur les langages rationnels des résultats plus précis et automatisables que sur les langages algébriques, ce qui justifie cette séparation.

5.1 Séries génératrices

Schématiquement, une série génératrice est une fonction dont le développement en série contient des informations de cardinalité sur des parties d'une famille d'objets. Une série génératrice est déduite de relations au sein de cette famille traduisibles en un système d'équations fonctionnelles, et dont la série génératrice est solution.
On donne ici de la série génératrice une définition un peu plus générale que la définition habituelle, de façon à rendre plus naturel le passage aux séries multivariées qui sont au centre de cette étude. Un lecteur impatient pourra cependant, sans perte d'information, substituer la taille usuelle sur les structures combinatoires à la notion plus générale de paramètre. On trouvera dans le livre de Wilf[27] une approche plus complète de cet outil.

5.1.1 Définitions

Définition 3 (fonction paramètre) :
Soit $\mathcal{A}$ une famille d'objets combinatoires.
On appelle paramètre une fonction de $\mathcal{A}$ dans $\mathbb{N}$.

Par extension, on appelera paramètre p d'un objet $o \in \mathcal{A}$ sa valuation $p(o)$ par p fonction paramètre.
Soit $\mathcal{A}$ et $\mathcal{B}$ deux ensembles d'objets combinatoires.

Définition 4 (Paramètre hérité additivement) :
Un paramètre p est dit hérité additivement si son extension sur $\mathcal{A} * \mathcal{B}$ est telle que :

$$\forall (a, b) \in \mathcal{A} * \mathcal{B}, p((a, b)) = p(a) + p(b)$$

Remarque : Dans toutes les classes d'intérêt en combinatoire, la taille de l'objet est un paramètre hérité additivement.
De plus, la relation r_p qui associe deux objets $(a, b) \in \mathcal{A}^2$ ssi $p(a) = p(b)$ est un relation d'équivalence. On qualifiera r_p de relation d'équivalence paramétrique.

Soit $\mathcal{A}$ une famille d'objets combinatoires.
Soit $(\mathcal{A}_i)_{i\geq 0}$ une partition de A définie par une relation d'équivalence paramétrique r_p ordonnée selon p.
Enfin, soit $a_i = |\mathcal{A}_i|, \forall i \geq 0$.

Définition 5 (Série génératrice ordinaire) :
On appelle série génératrice ordinaire de $\mathcal{A}$ la série formelle $A(z)$ telle que :

$$A(z) = \sum_{n\geq 0}^{\infty} a_n z^n = \sum_{o\in\mathcal{A}} z^{p(o)}$$

Plus simplement, la série génératrice ordinaire d'une classe $\mathcal{A}$ est la limite d'un polynôme dont le coefficient a_k est le nombre d'objets de paramètre égal à k.
En pratique, le paramètre étudié par les séries génératrices ordinaires est le plus souvent la taille. Les a_n sont alors les nombres d'objets de $\mathcal{A}$ de taille n. On parle alors de série génératrice de dénombrement.

5.1.2 Extraction de coefficients

La beauté des séries génératrices réside dans le fait qu'on peut les voir comme une somme infinie d'informations dont on peut trouver une expression sans en connaître explicitement les constituants. Une fois une expression de cette somme déterminée par des méthodes sur lesquelles nous reviendrons, on peut vouloir isoler une information spécifique au sein de cette somme. On peut pour cela utiliser le développement de Taylor.

Définition 6 (Coefficient d'un série génératrice) :
Le n^{ieme} coefficient d'une série génératrice $A(z) = \sum_{n\geq 0}^{\infty} a_n z^n$, noté $[z^n]A(z)$ est le coefficient du terme de degrès n dans le développement de Taylor de $A(z)$ en $z = 0$. On a donc :

$$[z^0]A(z) = A(0)$$

$$[z^n]A(z) = \frac{1}{n!}\frac{\partial^n A}{\partial z}(0)$$

Malheureusement, dans le cas général, l'obtention du coefficient de degrès n dans le développement de Taylor par le calcul de la dérivée n^{ieme} est susceptible de prendre un temps polynomial sur n et, sauf dans des cas très spécifiques, de rendre impossible une étude *statique* de nombreux problèmes. Il est donc intéressant d'exploiter d'autres résultats et propriétés combinatoires pour extraire les coefficients d'une série génératrice.

Théorème 1 (Inversion de Lagrange) :
Soit $A(z) = \sum_{n\geq 0}^{\infty} a_n z^n$ une série génératrice telle que $A(z) = z\Phi(A(z))$, pour $\Phi(0) \neq 0$, alors :

$$[z^n]A(z) = \frac{1}{n}[u^{n-1}](\Phi(u))^n$$

Cette formule est particulièrement pratique dans l'étude des structures arborescentes. En effet, une grande partie de celles ci peuvent être spécifiées de la façon suivante :

> Un arbre est soit une feuille, soit un noeud duquel partent des fils.

Ce qui nous conduit à l'équation fonctionelle suivante sur la série génératrice $A(z)$ de ces d'arbres :

$$A(z) = z + zA(z)^{x_1} + \ldots + zA(z)^{x_k}$$

où $x_1 \ldots x_k$ sont les différents nombres de fils autorisés pour un noeud. L'inversion de Lagrange est alors applicable, pour $\Phi(u) = 1 + u^{x_1} + \ldots + u^{x_k}$.
On l'applique ici au cas des arbres binaires, aussi appelés arbres de Catalan, de séries génératrice $B(z)$, qui ont comme *polynome caractéristique* $\Phi(u) = 1 + u^2$.
On obtient alors en utilisant l'inversion de Lagrange :

$$[z^n]B(z) = \frac{1}{n}[u^{n-1}](1+u^2)^n$$

On remarque que les coefficients de degrès pairs sont nuls, et on obtient, en utilisant la formule du binome de Newton :

$$[z^{2n+1}]B(z) = \frac{1}{2n+1}[u^n](1+u)^{2n+1} = \frac{1}{2n+1}\binom{2n+1}{n} = \frac{2n!}{n!(n+1)!}$$

On retrouve ici le nombre de Catalan, qui est omniprésent en combinatoire.

5.1.3 *Algébre minimale des séries génératrices ordinaires*

On va décrire ici un ensemble minimal de relations pouvant être transposée des ensembles aux séries génératrices. On appelle parfois cet ensemble de relation un *dictionnaire*.
Soient $A(z)$, $B(z)$ et $C(z)$ les deux séries génératrices ordinaires de trois ensemble d'objets. combinatoires $\mathcal{A}$, $\mathcal{B}$ et $\mathcal{C}$.

Théorème 2 :

$$\mathcal{C} = \mathcal{A} \times \mathcal{B} \Rightarrow C(z) = A(z) * B(z)$$

Si $\mathcal{A}$ et $\mathcal{B}$ disjoints :

$$\mathcal{C} = \mathcal{A} \cup \mathcal{B} \Rightarrow C(z) = A(z) + B(z)$$

Preuve : *Soit $A(z) = \sum_n a_n z^n$, $B(z) = \sum_n b_n z^n$ et $C(z) = \sum_n c_n z^n$*

- $\mathcal{C} = \mathcal{A} \times \mathcal{B}$:
  Le paramètre étant hérité additivement, les objets de $\mathcal{C}$ de paramètre n sont les paires (a, b) dans $\mathcal{A} \times \mathcal{B}$ pour lesquelles la somme des paramètres pour a et b est égale à n.$\Rightarrow c_n = \sum_{i=0}^{n} a_i * b_{n-i}$
  $\Rightarrow A(z) * B(z) = \sum_n a_n z^n * \sum_n b_n z^n = \sum_n (\sum_{i=0}^{n} a_i * b_{n-i} z^n) = C(z)$

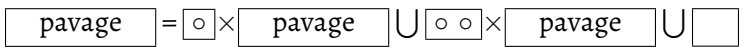
- $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$:
  $\mathcal{A}$ et $\mathcal{B}$ disjoints $\Rightarrow c_n = a_n + b_n$
  $\Rightarrow A(z) + B(z) = \sum_n (a_n + b_n) z^n = \sum_n c_z^n = C(z)$

Théorème 3 :

$$\mathcal{C} = Sequence(\mathcal{A}) \Rightarrow C(z) = \bigcup_{n \geq 0} A(z)^n = \frac{1}{1 - A(z)}$$

On peut donc transformer une décomposition éventuellement récursive d'un ensemble de structures combinatoires en système d'équations fonctionelles contraignant très fortement sa série génératrice.

On considère l'ensemble des pavages d'un segment $[1, n]$ avec des dominos $\boxed{\circ}$ et $\boxed{\circ\ \circ}$ de tailles respectives 1 et 2. On obtient facilement la décomposition suivante, où $\square$ désigne le pavage du segment de taille nulle :



On transpose alors cette décomposition en une relation caractérisant la série génératrice de ces pavages :

$$P(z) = zP(z) + z^2P(z) + 1$$

$$\Rightarrow P(z) = zP(z) + z^2P(z) + 1 = \frac{1}{1 - z - z^2}$$

En outre, on peut remarquer une égalité entre le nombre de pavages de $[1, n]$ et le $n - 1$ième nombre de Fibonacci. En effet :

$$P(z) = zP(z) + z^2P(z) + 1$$

$$\sum_{n\geq 0} p_n z^n = \sum_{n\geq 0} p_n z^{n+1} + \sum_{n\geq 0} p_n z^{n+2} + 1$$

$$[z^0]\sum_{n\geq 0} p_n z^n = [z^0]\sum_{n\geq 0} p_n z^{n+1} + [z^0]\sum_{n\geq 0} p_n z^{n+2} + 1 \Rightarrow p_0 = 1$$

$$[z^1]\sum_{n\geq 0} p_n z^n = [z^1]\sum_{n\geq 0} p_n z^{n+1} + [z^1]\sum_{n\geq 0} p_n z^{n+2} \Rightarrow p_1 = p_0 = 1$$

Enfin, pour $n \geq 2$:

$$[z^n]\sum_{n\geq 0} p_n z^n = [z^n]\sum_{n\geq 0} p_n z^{n+1} + [z^n]\sum_{n\geq 0} p_n z^{n+2} \Rightarrow p_n = p_{n-1} + p_{n-2}$$

5.1.4 Séries multivariées

Imaginons maintenant une classification des objets de $\mathcal{A}$ établie selon plusieurs paramètres. Par exemple, on peut vouloir compter les mots d'un langage selon les nombres d'occurences des différents caractères.
On introduit donc la notion de série génératrice multivariée, qui compte les objets de $\mathcal{A}$ selon plusieurs paramètres $p_1, \ldots, p_k$.
Soit $a_{d_1,\cdots,d_k}$ le nombre d'objets de $\mathcal{A}$ ayant d_i pour paramètre p_i(formellement, les nombres d'objets o tels que $p_i(o) = d_i$).

Définition 7 (Série génératrice multivariée) :
On appelle série génératrice multivariée de $\mathcal{A}$ la série formelle $A(u_1, \cdots, u_k)$ telle que :

$$A(u_1, \cdots, u_k) = \sum_{n\geq 0}^{\infty} a_{d_1,\cdots,d_k} {u_1}^{d_1} \cdots {u_k}^{d_k} = \sum_{o\in\mathcal{A}} {u_1}^{p_1(o)} \cdots {u_k}^{p_k(o)}$$

Les propriétés d'union et produit des séries génératrices sont conservées dans l'univers multivarié, car les paramètres sont additifs.
On va utiliser ce type de série pour compter les mots de $\{0, 1\}^*$ décrit par l'expression régulière $r = (0(0)^*1)^*$ à la fois selon la taille et le nombre d'occurences de la lettre 1.
Pour des raisons qu'on expliquera plus tard, la décomposition récursive *classique* des expressions

régulières peut ici être transposée en relation entre les séries génératrices, l'étoile de Kleene étant transposée en séquence.
On notera $L_r(z,u)$ la série génératrice bivariée du langage d'une expression régulière r.
On attribuera au paramètre *taille*(rep. *nombres occurences de* 1) la variable z(resp. u) .
Enfin, on notera $|w|_c$ le nombres d'occurences de c dans w.

$$L_0(z,u) = z^{|0|}u^{|0|_0} = zu$$

$$L_{(0)^*}(z,u) = Seq(L_0) = \frac{1}{1-zu}$$

$$L_{0(0)^*}(z,u) = L_0 L_{(0)^*} = \frac{zu}{1-zu}$$

$$L_1(z,u) = z^{|1|}u^{|1|_0} = z$$

$$L_{0(0)^*1}(z,u) = L_{0(0)^*}L_1 = \frac{z^2u}{1-zu}$$

$$L_r(z,u) = Seq(L_{0(0)^*1}) = \frac{1}{1-\frac{z^2u}{1-zu}}$$

5.2 Méthodologie DSV

Comme on l'a vu dans l'exemple précédent, les relations sur les séries génératrices peuvent êtres déduites de règles de construction de langages. La méthodologie DSV, formulée par Shützenberger, exploite cette propriété en établissant une bijection entre les objets combinatoires étudiés et les mots d'un langage décrit par une grammaire non ambiguë.

Définition 8 (Série génératrice de langage) :
On appelle série génératrice d'un langage $\mathcal{L}$ sur un alphabet $\{s_1,\dots,s_k\}$ la série formelle $L(z,x_1,\dots,x_k)$ telle que :

$$L(z,x_1,\dots,x_k) = \sum_{\omega\in\mathcal{L}} z^{|\omega|}x_1^{|\omega|_{s_1}}\dots x_k^{|\omega|_{s_k}} = \sum_{i_1+\dots+i_k=n} l_{n,i_1,\dots,i_k} z^n x_1^{i_1}\dots x_k^{i_k}$$

Où $|\omega|_s$ est le nombre d'occurences de s dans ω et $l_{n,i_1,\dots,i_k}$ le nombre de mots du langage de taille n ayant i_j occurences de s_j, pour $j\in[1,k]$.

On en déduit un système d'équation contraignant les séries génératrices des langages des non terminaux de la grammaire.

Mettons en oeuvre cette méthodologie dans l'étude des arbres de unaires-binaires. Partant d'un noeud initial appelé *racine*, un arbre unaire-binaire est :

- Soit une feuille.
- Soit un noeud ayant un unique fils unaires-binaires.
- Soit un noeud ayant deux fils unaires-binaires.

On trouve une bijection entre les arbres unaires binaires de taille $n+1$ et les mots de Motzkin de taille n.
Les mots de Motzkin sont les mots w sur l'alphabet $\{a,b,c\}$ tels que $|w|_a = |w|_b$ et, pour tout v préfixe de w, $|w|_a \leq |w|_b$.

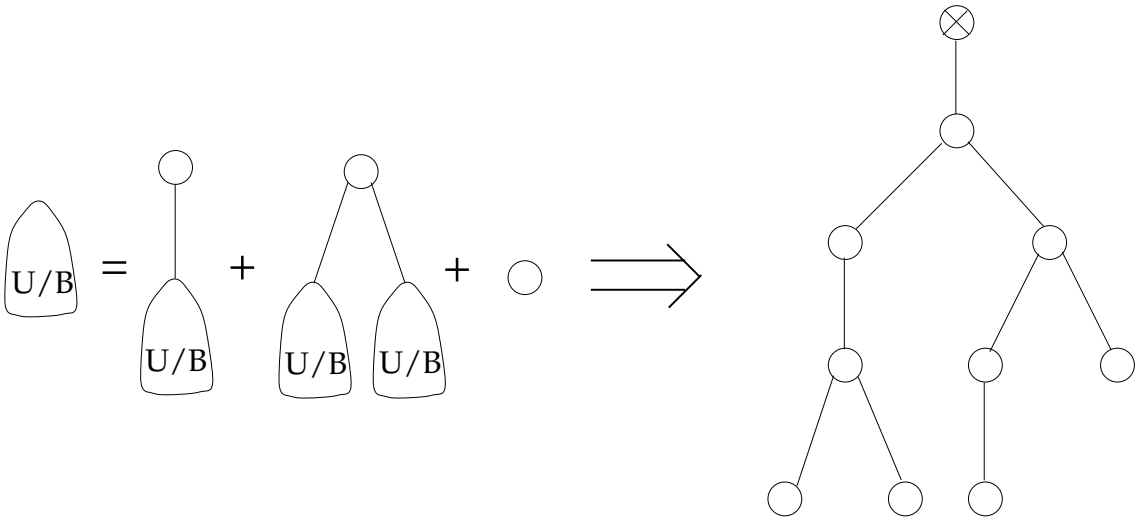


Fig. 5.1 : Un arbre unaire-binaire

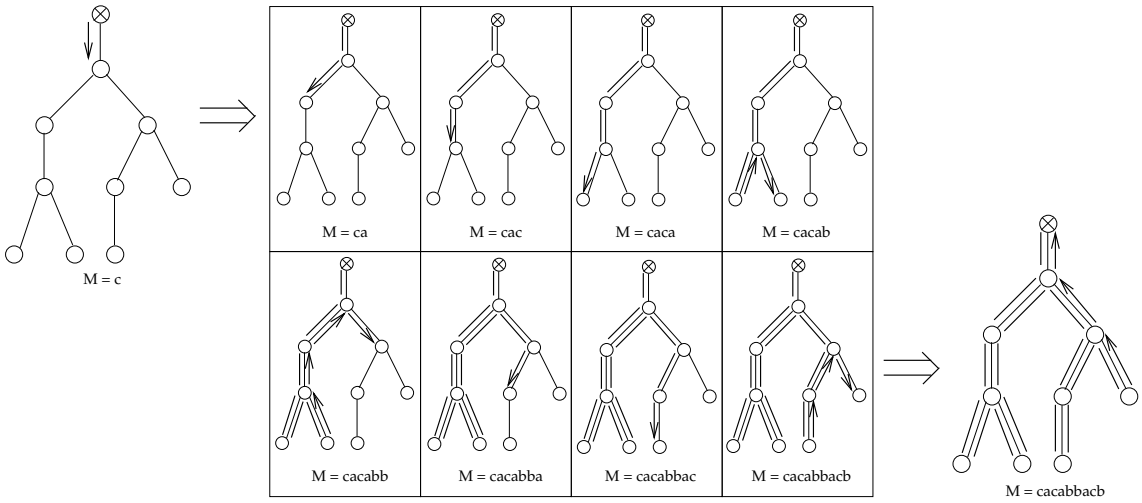


Fig. 5.2 : Transformation arbres unaires-binaires/mots de Motzkin

- Tout arbre unaire-binaire peut se réécrire en mot de Motzkin :
  On effectue un parcours en profondeur de l'arbre. Quand on traverse dans le sens descendant :
  - Un fils gauche, on écrit la lettre a.
  - Un fils droit, on écrit un b.
  - Un fils unique, on écrit un c.

  Les mots w ainsi générés séquentiellement au cours du parcours de l'arbre de taille $n+1$ sont bien des mots de Motzkin de taille n car :
  - Dans un arbre, il y a autant de fils gauches que de fils droits
    $\Rightarrow |w|_a = |w|_b$

- Le parcours en profondeur de l'arbre rencontre toujours le fils gauche d'un sous arbre binaire avant d'en rencontrer le fils droit ⇒ pour tout v préfixe de w, $|w|_a \leq |w|_b$
- Le nombre d'arètes dans un arbre de taille $n+1$ est n(Par récurrrence).

De plus, deux arbres différents donnent deux mots différents (trivial).

- Tout mot de Motzkin peut être transformé en un arbre unaire binaire :
  On lit le mot w de taille, en partant d'une racine vide :
  - Sur un a : On créé un branchement binaire ayant pour origine le noeud sélectionné et on sélectionne le fils gauche.
  - Sur un b : On se déplace sur le premier fils droit disponible à droite du noeud selectionné
  - Sur un c : On créé un branchement unaire et on sélectionne son extremité.

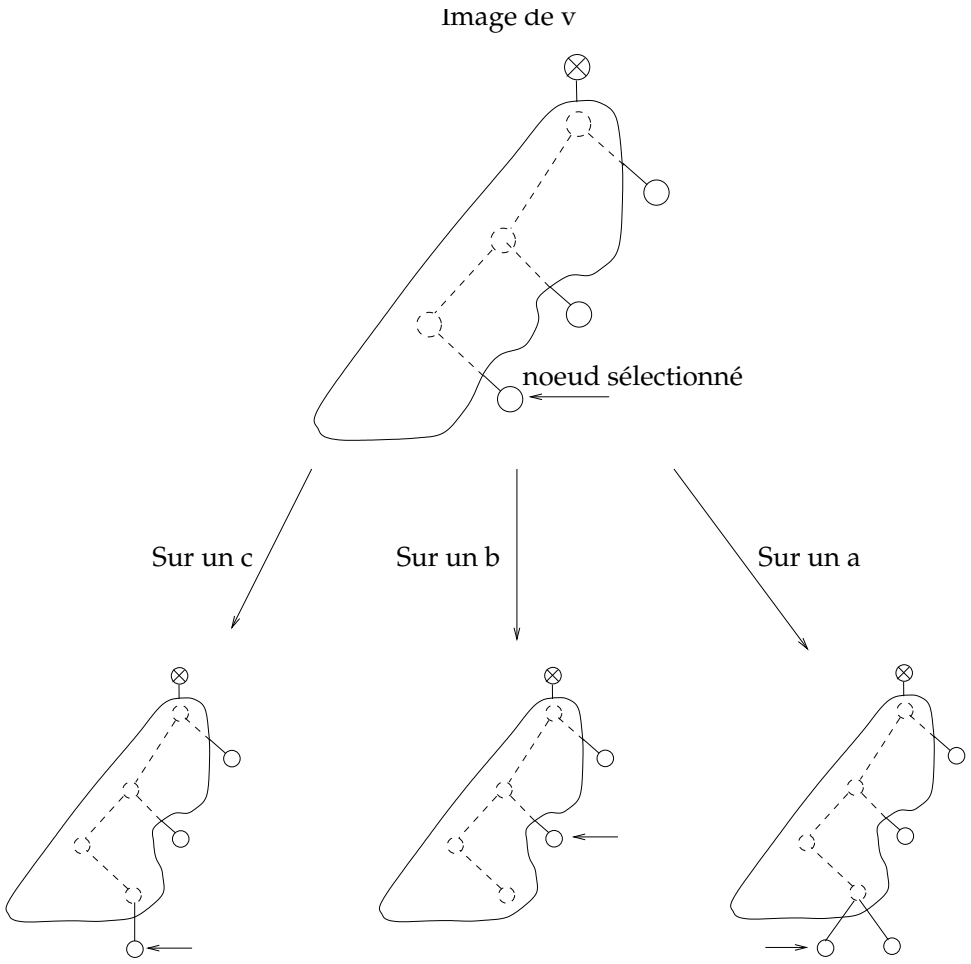


Fig. 5.3 : Deux arbres issus de mots ayant un plus long préfixe commun v sont distincts

Cet arbre est bien unaire-binaire, car tous les branchements ajoutés sont d'arité 1 ou 2. De plus, chaque a créé un fils droit disponible, donc $|w|_a \leq |w|_b$ implique qu'on pourra toujours trouver un fils droit disponible ⇒ on pourra toujours appliquer la règle c.
De plus, ces arbres ont bien $n+1$ noeuds, car chaque règle ajoute un noeud et on part d'une racine initiale.

Enfin deux mots différents donnent deux arbres différents car la lecture de deux caractères différents en un même noeud p (ce qui arrive forcément, ne serait ce qu'au noeud initial), donne des arbres issus de p déjà distincts au niveau des fils directs.

On a donc montré une bijection entre les mots de Motzkin et les arbres binaires unaires.
D'autre part, on sait que les mots de Motzkin sont engendrés par la grammaire non contextuelle non ambiguë suivante :

$$M \rightarrow aMbM \,|\, cM \,|\, \epsilon$$

Cette grammaire étant non ambiguë, on peut transformer ses règles de production en relations fonctionelle sur $M(z, x_a, x_b, x_c)$, série génératrice des mots issus du non terminal M.

$$\begin{aligned} M(z,x_a,x_b,x_c) &= zx_aM(z,x_a,x_b,x_c)zx_bM(z,x_a,x_b,x_c) + zx_cM(z,x_a,x_b,x_c) + 1 \\ 0 &= z^2x_ax_bM(z,x_a,x_b,x_c)^2 + (zx_c-1)M(z,x_a,x_b,x_c) + 1 \end{aligned}$$

En résolvant le système ci dessus pour $M(z,1,1,1) = M(z)$, on trouve deux candidats pour $M(z)$:

$$\begin{aligned} M_1(z) &= \frac{1-z+\sqrt{1-2z-3z^2}}{2z^2} \\ M_2(z) &= \frac{1-z-\sqrt{1-2z-3z^2}}{2z^2} \end{aligned}$$

$M(z)$ est une série de dénombrement, donc le coefficient de degrès n de son développement doit être le nombre de chemin de Motzkin de taille n. Or, le développement de $M_1(z)$ fait apparaitre des coefficients négatifs :

$$M_1(z) = \frac{1}{z^2} - \frac{1}{z} - 1 - z - 2z^2 - 4z^3 - 9z^4 + O(z^5)$$
$$\Rightarrow M(z) = M_2(z)$$

On obtient alors aisément la série génératrice des arbres unaires-binaires. En effet, les mots de Motzkin de taille n étant en bijection avec les arbres unaires-binaires de taille $n+1$, on a la relation suivante entre les séries génératrices $M(z)$ et $UB(z)$:

$$[z^n]M(z) = [z^{n+1}]UB(z) \Rightarrow M(z) = \sum_{n\geq 0} m_n z^n et UB(z) = ub_0 + \sum_{n\geq 0} m_n z^{n+1}$$

$$\Rightarrow UB(z) = ub_0 + z\sum_{n\geq 0} m_n z^n = ub_0 + zM(z)$$

Où ub_0 est le nombre d'arbres de taille 0. Or $ub_0 = 0$, car les arbres unaires-binaires partent nécessairement d'une racine.

$$\Rightarrow UB(z) = \frac{1-z-\sqrt{1-2z-3z^2}}{2z}$$

Cette méthodologie ne permet pas d'étudier toutes les familles d'objets d'intérêt en combinatoire. On caractérise cependant deux grandes familles d'objets pour lesquelles on obtient des séries génératrices de langages remarquables, les langages rationnels et algébriques.

5.2.1 Langages rationnels

Définition 9 (Fraction rationelle) :
On appelle fraction rationnelle une fonction $F(z)$ telle que :

$$F(z) = \frac{A(z)}{B(z)}$$

où $A(z)$ et $B(z)$ sont des polynômes en $z \in \mathbb{C}$ à coefficient dans $\mathbb{N}$ ou $\mathbb{R}$

Théorème 4 :
Les séries génératrices des langages rationnels sont des fractions rationnelles.

Preuve (Inspirée de Chomsky Schützenberger) :
Soit $\mathcal{L}$ un langage rationnel.
D'après le théorème de Kleene : Rat = Reg. Donc il existe un automate fini $A = (Q, V, q_i, F, \delta)$ déterministe minimal reconnaissant exactement $\mathcal{L}$. On rappelle que δ est la fonction de transition de l'automate, qui à un état q et une lettre c associe l'état successeur q' de q sur lecture de c .
Soit $\mathcal{L}_q$ le langage des mots reconnus par A à partir d'un état q, c'est à dire l'ensemble des mots $w \in A^*$ tels que $q \rightsquigarrow_w f \in F$. On a la relation suivante :

$$\mathcal{L}_q = \bigcup_{c \in V} \{c\}.\mathcal{L}_{\delta(q,c)} \quad (+\varepsilon)_{Si\ q \in F}$$

De plus, cette union est disjointe, car les préfixes c des différents ensembles constitutifs de $\mathcal{L}_q$ sont distincts. Donc on peut utiliser le dictionnaire sur les séries génératrices ordinaires pour transposer les relations entre les langages $\mathcal{L}_q$ en relations sur les séries génératrices $L_q(z)$.

$$L_q(z) = \sum_{c \in V} z L_{\delta(q,c)}(z) \quad (+1)_{Si\ q \in F}$$

On obtient donc un système d'équations linéaires qui admet la représentation matricielle R suivante :

$$R = (r_{i,j})_{\substack{1 \le i \le |Q| \\ 1 \le j \le |Q|+1}} = \left(\begin{array}{cccc|c} z\delta_{1,1} - 1 & z\delta_{1,2} & \cdots & z\delta_{1,|Q|} & \delta_{1,F} \\ z\delta_{2,1} & z\delta_{2,2} - 1 & \cdots & z\delta_{2,|Q|} & \delta_{2,F} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z\delta_{|Q|,1} & z\delta_{|Q|,2} & \cdots & z\delta_{|Q|,|Q|} - 1 & \delta_{|Q|,F} \end{array}\right)$$

$$\delta_{i,j} = \begin{cases} 1 & si\ \exists c\ tq\ q_i \to_c q_j \\ 0 & sinon \end{cases} \quad et \quad \delta_{i,F} = \begin{cases} 1 & si\ q_i \in F \\ 0 & sinon \end{cases}$$

Le carré supérieur gauche de cette matrice est diagonalisable, car seuls les termes $r_{i,i},\ 1 \le i \le |Q|$ ont un coefficient de degrès 0 non nul. Il n'existe donc pas de combinaison linéaire des vecteurs lignes $\{r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{|Q|}\}$ qui soit égale à r_i.
Une fois le carré supérieur gauche de la matrice R diagonalisé, par une méthode du type pivot de Gauss, on obtient une matrice S telle que :

$$S = \left(\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & L_{q_1}(z) \\ 0 & 1 & \cdots & 0 & L_{q_2}(z) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & L_{q_{|Q|}}(z) \end{array}\right)$$

La série génératrice $L(z)$ du langage $\mathcal{L}$ est alors $L_{q_i}(z)$ (On rappelle que q_i est l'état initial de A)
Celle ci est une fraction rationnelle car les seules opérations mise en jeux dans la diagonalisation d'une matrice sont l'addition et la soustraction de vecteurs, ainsi que la multiplication et la division par des polynomes, opérations par lesquelles les fractions rationelles sont closes. $L(z) = L_{q_i}(z)$, série génératrice d'un langage rationnel $\mathcal{L}$ est donc une fraction rationelle.

5.2.2 Langages algébriques

Définition 10 (Fonction Algébrique) :
Une fonction algébrique est une fonction f d'une variable réelle ou complexe pour laquelle il existe $P(u,z)$ polynôme tel que :

$$P(f(z),z)=0$$

Définition 11 (alt. Fonction Algébrique) :
Une fonction algébrique est la projection d'une variété algébrique projective, c'est à dire une composante y_i d'un vecteur solution du système d'équations :

$$\begin{cases} y_1(z) &= \Phi_1(z,y_1(z),\ldots,y_n(z)) \\ y_2(z) &= \Phi_2(z,y_1(z),\ldots,y_n(z)) \\ \vdots & \vdots \quad \vdots \\ y_n(z) &= \Phi_n(z,y_1(z),\ldots,y_n(z)) \end{cases}$$

où $\Phi_1,\Phi_2,\ldots,\Phi_n$ sont des polynômes.

Théorème 5 :
Les séries génératrices des langages algébriques sont des fonctions algébriques.

Preuve :
Un langage $\mathcal{L}$ est algébrique $\Leftrightarrow$ Il existe une grammaire non contextuelle G non ambiguë qui engendre exactement les mots de $\mathcal{L}$.
On s'interesse aux règles de réécriture de cette grammaire, mise en forme normale de Chomsky.(Un passage par une telle forme n'est pas nécessaire, mais facilite la lecture de la preuve) On rappelle que la forme normale de Chomsky d'une grammaire ne contient que des règles r du type :

$$r=\begin{cases} & A\to B.C \\ ou & \\ & A\to B|C \\ ou & \\ & A\to c \end{cases}$$

Où A,B et C sont des non terminaux, et a un caractère du vocabulaire terminal. La mise en forme normale de Chomsky d'une grammaire non ambiguë est non ambiguë, donc on peut transposer les règles en relations sur les langages engendrés par les non terminaux, puis en déduire des relations sur les séries génératrices ordinaires. Soit $\mathcal{L}_\mathcal{A}$ le langage des mots issus du non terminal A et $L_A(z)$ sa série génératrice de dénombrement :

$$\begin{array}{lcll} A\to B.C & \Rightarrow & \mathcal{L}_\mathcal{A}=\mathcal{L}_\mathcal{B}\times\mathcal{L}_\mathcal{C} & \Rightarrow \quad L_A(z)=L_B(z)*L_C(z) \\ A\to B|C & \Rightarrow & \mathcal{L}_\mathcal{A}=\mathcal{L}_\mathcal{B}\bigcup\mathcal{L}_\mathcal{C} & \Rightarrow \quad L_A(z)=L_B(z)+L_C(z) \\ A\to c & \Rightarrow & \mathcal{L}_\mathcal{A}=\{c\} & \Rightarrow \quad L_A(z)=z \end{array}$$

On obtient ainsi un système d'équations de la forme $L_i(z)=\Phi_A(z,L_1(z),\ldots,L_n(z))$ où n est le nombre de non terminaux. La série génératrice de ce langage est la série associée au non terminal initial de la grammaire. Elle est algébrique d'après la définition alternative des fonctions algébriques.

5.3 Techniques de combinatoire asymptotique

On a présenté jusqu'ici un quelques techniques **exactes** pour l'extraction des coefficients d'une série génératrice. Celles ci, quand elles peuvent être appliquées, profitent de l'apparition d'un motif particulier dans l'expression de la série génératrice. Un tel traitement *symbolique* de la série

génératrice n'est donc que très rarement possible.
Dans baucoup de cas, l'obtention des coefficients nécessite l'execution d'algorithmes de complexités polynomiales, voire exponentielles sur l'indice du coefficient recherché. On relâche donc la contrainte d'un calcul exact pour s'intéresser au *comportement asymptotique* des coefficients. De même qu'on a pu, dans le cas des séries génératrices, trouver la limite d'une suite de polynômes dont on ne connaissait pas explicitement les coefficients, on arrive dans bien des cas à déterminer le comportement asymptotique des coefficients d'une série génératrice.
Comme on l'a vu précedemment, on peut, grace à la méthodologie DSV, exploiter le système d'équations fonctionnelles induit par les règles de constructions non ambiguës d'un langage pour en déterminer la série génératrice. On distingue les séries génératrices obtenues selon la classe à laquelle appartient leur langage associé dans la hierarchie de Chomsky.

Il existe un lien étroit entre le comportement d'une fonction analytique aux alentours de sa singularité de plus petit module et le comportement asymptotique des coefficients de son développement en série. Pour comprendre cette assertion, nous allons introduire quelques concepts d'analyse complexe.

5.3.1 Intégrale de contours et coefficients de séries

Définition 12 (Fonction Analytique) :
Une fonction f est analytique sur un domaine Ω si, en tout point de Ω, elle admet un développement en série entière convergeant localement.

Définition 13 (Fonction Méromorphe) :
Soient g et h deux fonctions analytiques sur $\Omega \in \mathbb{C}$, alors leur quotient :

$$f(z) = \frac{g(z)}{h(z)}$$

est analytique sur Ω sauf en les points α où $h(\alpha) = 0$, appelés **pôles** *de f.*
On dit alors que $f(z)$ est méromorphe en $z = \alpha$.

Alternativement, une fonction f méromorphe en $z = \alpha$ est une fonction qui admet un développement de la forme

$$f(z) = \sum_{n \geq -M} f_n (z - \alpha)^n$$

dans un voisinage de α ne contenant pas α.
Si $f_{-M} \neq 0$, alors M est l'**ordre** du pôle α.
Enfin, si $M \geq 1$, f_{-1} est appelé le **résidu** de $f(z)$ en $z = \alpha$, qu'on note $Res[f(z), z = \alpha]$.

Théorème 6 (Théorème du résidu de Cauchy) :
Soit Ω une région de $\mathbb{C}$ simplement connexe.
Soit Γ une courbe fermée orientée dans le sens positif inclue dans Ω.
Soit $h(z)$ méromorphe sur Ω, sauf en un nombre fini de points isolé $\alpha_1, \alpha_2, \ldots, \alpha_k$.
Alors on a l'équation suivante :

$$Res[f(z), z = \alpha] = \frac{1}{2\pi i} \int_\Gamma f(z) dz$$

On en déduit directement une expression du coefficient de degrès n dans le développement autours de 0. En effet, il existe un cercle Ω centré en 0 de rayon suffisamment petit pour ne contenir

aucune singularité de $f(z)$. On choisit alors d'étudier $\frac{f(z)}{z^{n+1}}$, qui admet une singularité en 0 pour n suffisament grand.

$$\frac{1}{2\pi i}\int_{\Gamma}\frac{f(z)}{z^n+1}dz = Res[\frac{f(z)}{z^n+1}, z=0] = f_n$$

On obtient donc le coefficient de degrès n dans le développement de Taylor de la série génératrice autours de 0, qui est la donnée d'intérêt en combinatoire.

Une telle méthode ne peut pas servir de base à un traitement automatisé des séries génératice en vue d'en extraire les coefficients, à cause notamment de la difficulté d'intégrer des fonctions sur un contours complexe[1]. Cependant, on voit désormais que les singularités(ici, le pôles), d'une fonction méromorphe vont intervenir dans les comportements des coefficients de la série génératrice.

Le dogme central de l'analyse de singularités est le suivant :

- *La singularité de plus petit module d'une série génératrice détermine le comportement exponentiel des coefficients.*
- *Le type de singularité détermine le comportement sous exponentiel.*
- *Les constantes multiplicatives du comportement asymptotique sont déduites du comportement de la série au voisinage de la singularité dominante.*

Pour le troisième point, on utilise une propriété[2] vérifiée par les singularités des séries génératrices rationelles ou algébriques pour utiliser le transfert :

$$f(z) \underset{z\to\rho}{\sim} g(z) \Rightarrow [z^n]f(z) \underset{z\to\infty}{\sim} [z^n]g(z)$$

Où ρ est la singularité dominante, dont on verra par la suite le rôle prédominant, et $g(z)$ un équivalent de z aux alentours de ρ, qui ne conserve que les propriétés *pathologiques* de $f(z)$ en ρ.

5.3.2 Ordre exponentiel et singularités

Définition 14 (Ordre exponentiel) :
Soit $\{a_n\}$ une suite.
On appelle ordre exponentiel de $\{a_n\}$ la fonction K^n telle que :

$$a_n \asymp K^n$$

C'est à dire $|a_n| = \vartheta(n)K^n$ où $\vartheta(n)$ est un facteur subexponentiel :

$$\lim_{n\to\infty}\sup|\vartheta(n)|^{\frac{1}{n}} = 1$$

Définition 15 (Singularité) :
On appelle singularité d'une fonction f analytique sur Ω un ouvert de $\mathbb{C}$ un point α sur la frontière de Ω tel qu'on ne peut pas prolonger f en α par une fonction analytique.

[1] En pratique, le théorème des résidus est perçu par les mathématicien, dont Cauchy, commme un outil d'intégration complexe, ce problème étant *à priori* plus difficile que l'extraction des coefficients d'une série.

[2] Fonction continuable dans un *camembert* centré sur la singularité, et d'angle inférieur à π, ce qui permet l'application du théorème de Darboux.

En pratique, les singularités que nous rencontrerons sont les racines des dénominateurs, appelées pôles et les zéros des opérandes de racines, qualifiées d'algébriques.

Définition 16 (Singularité dominante) :
Soit f une fonction analytique en 0. On appelle singularité dominante la singularité de f de plus petit module.

Théorème 7 (Ordre exponentielle des coefficients d'une série) :
Soit $f(z)$ une fonction analytique en 0 et ρ le module de la singularité dominante de f.
Alors :

$$a_n \asymp \left(\frac{1}{\rho}\right)^n$$

Il est intéressant de noter que le comportement exponentiel des coefficients d'une série génératrice ne dépend pas du type de singularité.

Théorème 8 (Théorème de Pringsheim) :
Une série entière à coefficients positifs et de rayon de convergence fini ρ a une de ses singularité dominantes sur le demi axe des réels positifs.

Ce théorème facilite grandement l'étude asymptotique des séries génératrices. En effet, les séries étudiées ayant des coefficients positifs, on pourra résoudre les équations determinant les pôles dans $\mathbb{R}^+$.

5.3.3 Asymptotique des séries génératrices rationelles

Théorème 9 (Expansion des fonctions rationelles) :
Soit $f(z) = \frac{P(z)}{Q(z)}$ une fraction rationelle analytique en 0.
Soit $\alpha_1, \dots, \alpha_m$ les pôles de $f(z)$ ($\Leftrightarrow$ les racines de $Q(z)$) de multiplicités $k_1, \dots, k_m$
Alors il existe m polynômes $\Pi_1(x), \dots, \Pi_m(x)$, de degrès $k_1 - 1, \dots, k_m - 1$ et $M \leq deg(P(z))$ tels que :

$$f_n \equiv [z^n]f(z) = \sum_{i=1}^{m} \Pi_i(n)\alpha_i^{-n} \quad , \forall n > M \tag{5.1}$$

Preuve : *On suppose que $deg(P) < deg(Q)$. On décompose alors $f(z)$ en éléments simples :*

$$f(z) = \sum_{i=1}^{m}\sum_{j=1}^{k_i} \frac{c_{\alpha_i,j}}{(z-\alpha_i)^j}$$

De plus :

$$[z^n]\sum_j f_j(z) = \sum_j [z^n]f_j(z)$$

Donc on peut étudier séparément les contributions aux coefficient de $f(z)$ de chacun de ces éléments simples. Celles ci sont de la forme suivante :

$$\begin{aligned}
[z^n]\tfrac{1}{(z-\alpha_i)^j} &= [z^n]\tfrac{1}{(-1)^{-j}\alpha_i^j(1-\frac{z}{\alpha_i})^j} \\
&= \tfrac{(-1)^j}{\alpha_i^j}[z^n]\tfrac{1}{(1-\frac{z}{\alpha_i})^j} \\
&= \tfrac{(-1)^j}{\alpha_i^j}\tbinom{n+j-1}{j-1}\alpha_i^{-n}
\end{aligned}$$

Le binomial pris comme un polynôme en n est de degrès $j-1$. On rappelle que j est borné par la multiplicité de la racine.

De plus, l'hypothèse que $deg(P) < deg(Q)$ peut être formulée sans perte de généralité, car dans le cas contraire, on ne peut certes pas décomposer en éléments simples, mais on peut soustraire à P ses termes de degrès $\geq deg(Q)$. On obtient alors P' tel que $deg(P') < deg(Q)$ qu'on peut décomposer en éléments simples. L'équation (5.1) *est donc valable pour $n > deg(P)$.*

On déduit immédiatement de l'équation (5.1) l'asymptotique des coefficients de la série. En effet, asymptotiquement, seul le terme $\Pi_i(n)\alpha_i^{-n}$ de plus petit α_i s'exprime, écrasant exponentiellement les autres termes.

Théorème 10 (Asymptotique des fonctions rationelles) :
Soit $f(z) = \frac{P(z)}{Q(z)}$ une fraction rationelle analytique en 0.
Soit α le pôle de $f(z)$ de plus petit module de multiplicité k et :

$$f(z) \underset{z\to\alpha}{\sim} \frac{c}{(1-\frac{z}{\alpha})^k}, k \in \mathbb{N}$$

Alors :

$$[z^n]f(z) \sim c\alpha^n \frac{n^{k-1}}{(k-1)!} \tag{5.2}$$

On en déduit une méthodologie pour l'asymptotique des coefficients d'une série rationelle f:

- Calculer les pôles de $f(z)$.
- Déterminer un équivalent asymptotique de f au voisinnage de son pôle de plus petit module.
- En déduire grace à (5.2) un équivalent pour les coefficients de f.

Nous utilisons cette méthodologie pour calculer le nombre de combinaisons de pièces de 1,2,5,10,20 et 50 cents d'euro dont la somme est n cents :

$$n = 1k_1 + 2k_2 + 5k_5 + 10k_{10} + 20k_{20} + 50k_{50}$$

$$M(z) = \sum_{k_i, i\in\{1,2,5,10,20,50\}} z^{\sum_i ik_i} = \sum_{k_1} z \sum_{k_2} z^2 \sum_{k_5} z^5 \sum_{k_{10}} z^{10} \sum_{k_{20}} z^{20} \sum_{k_{50}} z^{50}$$

$$\Rightarrow M(z) = \frac{1}{(1-z)(1-z^2)(1-z^5)(1-z^{10})(1-z^{20})(1-z^{50})}$$

Considérons les pôles : On a un pôle en $z = 1$ de multiplicité 6. D'après le théorème de Pringsheim, on peut se limiter aux solutions rélles positives. En outre, tous les pôles sont ici des racines de l'unité. Donc on peut, sans Pringsheim, en déduire qu'une singularité de plus petit module est bien présente sur $\mathbb{R}^+$. En outre, on utilise la petite astuce de calcul suivante :

$$(1-z^k) = (1-z)(\sum_{i=0}^{k-1} z^k)$$

Alors :

$$M(z) = \frac{1}{(1-z)^6} \cdot \frac{1}{(1+z)(1+\ldots+z^4)(1+\ldots+z^9)(1+\ldots+z^{19})(1+\ldots+z^{49})}$$

$$M(z) \underset{z\to1}{\sim} \frac{1}{(1-z)^6} \frac{1}{1.2.5.10.20.50}$$

Or, d'après (5.2) :

$$[z^n]\frac{1}{(1-z)^6}.\frac{1}{10^5} \underset{n\to\infty}{\sim} \frac{n^5}{5!}.\frac{1}{10^5} = \frac{n^5}{12.10^6}$$

$$\Rightarrow [z^n]M(z) \underset{n\to\infty}{\sim} \frac{n^5}{12.10^6}(1+O(\frac{1}{n^3}))$$

Le facteur n^3 vient du deuxième pôle en $z=-1$ de multiplicité 3, qui donnera lieu à une croissance cubique.

5.3.4 *Asymptotique des séries génératrices algébriques*

On rappelle qu'une série algébrique f est racine d'un polynôme P en $f(z)$ et z. Donc ses singularités sont algébriques, c'est à dire qu'elles apparaissent dans la série génératrice sous la forme de facteurs $(1-z/\alpha)^s$, où $s \in \mathbb{R}-\mathbb{N}$.

Théorème 11 (Echelle asymptotique de fonctions) :
Soit $f(z)$ une fonction telle que :

$$f(z) = \frac{1}{(1-\frac{z}{\alpha})^s} \quad s \in \mathbb{R}-\mathbb{N}$$

Alors :

$$[z^n]f(z) \underset{n\to\infty}{\sim} \alpha^n \frac{n^{s-1}}{\Gamma(s)}(1+O(\frac{1}{n}))$$

On rappelle que $\Gamma(s)$ est la fonction gamma de Euler, définie pour des x positifs de la façon suivante :

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}e^{-t}dt$$

et étendue à des x négatifs non entiers ou nuls par :

$$\Gamma(x)\Gamma(1-x) = \frac{\pi}{\sin(\pi x)}$$

En particulier :

$$\Gamma(1) = \int_0^{+\infty} e^{-t}dt = 1$$

$$\Gamma(x+1) = x\Gamma(x) \Rightarrow \Gamma(n+1\in\mathbb{N}) = n!$$

$$\Gamma(1/2) = \sqrt{\pi}$$

On montre en quoi ce *transfert* des coefficients de l'asymptotique d'une fonction à l'asymptotique des coefficients peut être utile en l'appliquant à l'étude de l'asymptotique des mots de Motzkin, qui sont déjà apparus lors de l'application de la méthodologie DSV à l'étude des arbre unaires/binaires et dont la série génératrice est :

$$M(z) = \frac{1-z-\sqrt{1-2z-3z^2}}{2z^2}$$

Cette fonction est analytique en 0 car prolongeable par 1 en 0. Les singularités sont donc de type algébrique, et dues à la présence de la racine carrée. Les deux racines du polynôme $1-2z-3z^2$

sont $1/3$ et -1. On ne s'intéresse donc qu'à $z = 1/3$, car c'est la racine qui a la plus grande contribution exponentielle à l'asymptotique de $[z^n]M(z)$. Or :

$$\begin{array}{rcl} M(z) & \underset{z\to 1/3}{\sim} & \frac{1-1/3}{2/9} + \frac{1-1/3-\sqrt{1+1/3}\sqrt{1-3z}}{2/9} \\ & = & 3 - 3\sqrt{3}\sqrt{1-3z} \\ & = & 3 - 3\sqrt{3}\frac{1}{(1-\frac{z}{1/3})^{-1/2}} \end{array}$$

Or, d'après le Théorème 11 :

$$[z^n]\frac{1}{(1-\frac{z}{1/3})^{-1/2}} \underset{n\to\infty}{\sim} \frac{3^n n^{3/2}}{\Gamma(-1/2)}(1+O(\frac{1}{n})) = -\frac{3^n n^{3/2}}{2\sqrt{\pi}}(1+O(\frac{1}{n}))$$

Car :

$$\begin{array}{lcl} \Gamma(-1/2)\Gamma(1+1/2) & = & \frac{\pi}{\sin(-\pi/2)} \\ \Gamma(-1/2) & = & \frac{\pi}{-1.1/2.\Gamma(1/2)} \\ & = & \frac{\pi}{-1.1/2.\sqrt{\pi}} \\ & = & -2\sqrt{\pi} \end{array}$$

Donc :

$$[z^n]M(z) \underset{n\to\infty}{\sim} [z^n]3 - 3\sqrt{3}\frac{1}{(1-\frac{z}{1/3})^{-1/2}}$$

$$[z^n]M(z) \underset{n\to\infty}{\sim} \frac{3\sqrt{3}}{2\sqrt{\pi}}3^n n^{3/2}(1+O(\frac{1}{n}))$$

On est donc capable de déterminer l'asymptotique des coefficients de séries génératrices algébriques, facteurs sous exponentiels et constantes comprises.

Armés de ces outils nous pouvons désormais nous attaquer à une modélisation par grammaire non contextuelle des structures secondaires d'ARN.

6. ETUDES HISTORIQUES DES STRUCTURES SECONDAIRES D'ARN

Dans cette partie, on établit un état de l'art des recherches sur les propriétés combinatoires des structures secondaires d'ARN.

Historiquement, le problème de la combinatoire des structures secondaires d'ARN est apparu lorsqu'on a commencé à s'intéresser au repliement de l'ARN. Les structures secondaires étant, comme on va le voir, en nombre exponentiellement moins élevé que les séquences sur les quatres bases, on a un nombre élevé de structures candidates pour une séquence donnée. Les premiers algorithmes fonctionnent donc sur le principe de minimisation de énérgie libre, minimisation qui ne peut être réalisée efficacement que si les objets manipulés disposent d'une structure récursive. Or, il n'existe qu'un pas entre la découverte d'une décomposition récursive et celle d'une série génératrice de dénombrement.

C'est pourquoi, dans une étude mathématique préliminaire à la conception d'un algorithme de repliement, M.S. Waterman obtient en [24] les premiers résultats combinatoires sur les structures secondaires d'ARN.
Cependant, cet article laisse quelques questions sans réponses. Par exemple, Waterman introduit la notion d'ordre des structures secondaires, un paramètre de complexité des structures secondaires, mais ne dénombre pas les structures d'une taille et d'un ordre donnés.

G. Viennot et M. Vauchaussade de Chaumont obtiennent en 1983 dans [2] les séries génératrices des structures d'un ordre donné, qui fait intervenir des polynômes définis par récurrence. Malheureusement, il semble impossible d'en déduire une forme générale pour les nombres s_{nk} de structures de taille n et d'ordre k.

En 2001, M. Nebel applique dans [16] un résultats obtenu précedemment sur le paramètre d'Orton Strahler pour déduire les comportements asymptotiques de certains paramètres, comme le nombre de renflements ou l'ordre.

On peut aussi noter les contributions de I. Hofacker et al., qui présentent en [13] des récurrences sur les nombres de structures ayant au moins c bases dans les boucles. Enfin, on peut trouver dans [19] un résultat de M. Régnier sur le nombre asymptotique de structures dans lesquelles les boucles sont constituées d'au moins b bases et les hélices d'au moins h paires de bases.

On rappelle la définition des structures secondaires établie dans la partie de ce mémoire consacrée au contexte biologique :

> *On appelle structure secondaire d'un ARN la structure planaire comportant les appariements Watson-Crick et Wobble, et ne comportant pas de pseudo-noeuds.*

Un telle définition est assez proche de celle d'un graphe non orienté, dans lequel les sommets sont les bases et les arètes les liaisons phosphodiesters et hydrogènes. C'est cette modélisation que choisit Watermann quand il étudie en 1978 les structures secondaires d'ARN.

6.1 Approche orientée graphes : Waterman 1978[24]

6.1.1 Définition des structures secondaires

Définition 17 (Structures secondaires de Waterman) :
Une structure secondaire est un graphe composé de sommets $\{1, \ldots, n\}$ *et doté d'une matrice d'adjacence* $A = (a_{ij})$ *telle que :*

(i) $a_{i\,i+1} = 1, \quad \forall i \in [1, n-1]$

(ii) $\forall i \in [1, n]$*, il existe au plus un* $j \neq i \pm 1$ *tel que* $a_{ij} = 1$

(iii) Si $a_{ij} = 1$,$a_{kl} = 1$ *et* $i < k < j$*, alors* $i \leq l \leq j$

Remarque : Bien que non formulé explicitement dans la définition, le graphe est non orienté, donc $a_{ij} = a_{ji}, \quad \forall (i, j) \in [1, n]^2$
On peut traduire ces trois conditions dans le *langage courant,* où leur nécessité devient alors évidente :

(i) $\Leftrightarrow$ *La structure secondaire contient les liaisons de la structure primaire*(liaison phosphodiesters).

(ii) $\Leftrightarrow$ *Les triple hélices sont interdites.*
En effet, la présence d'une triple hélice implique l'appariement de trois bases deux à deux non consécutives dans la séquence primaire. Il existe alors une base i appariée à deux bases qui ne lui sont pas consécutives, donc la condition est violée.
$\Leftrightarrow$ *Les boucles terminales de taille* 0 *sont interdites.*
D'après la définition d'un appariement, deux bases ne peuvent être appariées que si elles ne sont pas adjacentes. Or une boucle est délimitée par une paire de base appariées, donc si cette boucle est de taille 0, alors elle est délimitée par deux bases adjacentes et appariées, ce qui est impossible.

(iii) $\Leftrightarrow$ *On ne tolère pas les pseudo-noeuds.*
En terme de graphe, un pseudo-noeuds est un couple d'appariements *chevauchants,* c'est à dire (i, j) et (k, l) tels que $i < k < j < l$, ce qui est rendu impossible par la condition (iii).

Définition 18 (Appariement) :
Soit $A = (a_{ij})$ *la matrice d'adjacence d'une structure secondaire. On dit que deux bases* i *et* j *sont appariées si* $a_{ij} = 1$ *et* $|i - j| \neq 1$

6.1.2 Série génératrice de dénombrement

Théorème 12 (Série génératrice des structures secondaires d'ARN) :
La série génératrice de dénombrement des structures secondaires d'ARN est $S(z)$ *telle que :*

$$S(z) = \sum_{n \geq 0} s_n z^n = \frac{z^2 - z + 1 - \sqrt{1 + z(z^3 - 2z^2 - z - 2)}}{2z^2}$$

où s_n *est le nombre de structures secondaires de taille* n*.*

Lemme 1 (Récurrence sur les nombres de Str. Sec.) :
Les nombres s_n *de structures secondaires de taille* n *obéissent à la récurrence suivante :*

$$s_{n+1} = s_n + \sum_{i=1}^{n-2} s_i s_{n-i-1} \qquad s_0 = 1$$

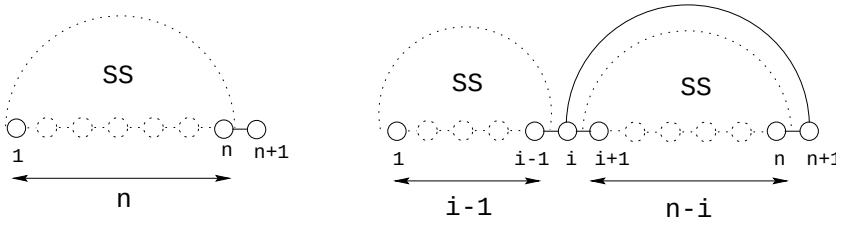


Fig. 6.1 : Décomposition des structures secondaire de taille $n+1$ en fonction de l'appariement de la $n+1$-ième base

Preuve (Lemme 1) : *Prenons une structure secondaire de $n+1$ bases et observons la $n+1$-ième :*

- $n+1$ *n'est pas appariée :*
  Donc les n bases $\{1,\ldots,n\}$ sont susceptibles de former n'importe quels appariements valides. Alors le nombre de str. sec. de taille $n+1$ ayant leur dernière base non appariée est égal au nombre de str. sec. de taille n = s_n

- $n+1$ *est appariée avec une base i :*
  Notons que $i < n$ car sinon il existe deux arète entre i et $n+1$.
  Alors les bases des l'intervalles $[1, i-1]$ et $[i+1, n]$ peuvent former n'importe quelles structures secondaires de tailles $i-1$ et $n-i$. De plus, pour deux couples distincts de structures secondaires de ces tailles, la structure secondaire composée obtenue est distincte par comparaison des matrices d'adjacence. Donc les structures secondaires dans lesquelles n est apparié à j sont au nombre de $s_{i-1}s_{n-i}$.
  Enfin, on remarque que deux str. sec. dans lesquelles les $n+1$-ièmes bases sont appariées avec des bases i et i', $i \neq i'$ ont des matrices de transitions différentes pour le coefficient $a_{i\,n+1}$, par la condition (ii) du théorème 17. Donc les ensembles de structures secondaires ayant leur $n+1$-ième base appariée avec des bases distinctes sont disjoints.

  Les deux cas de la décomposition couvrent des ensembles de str. sec. disjoints, car "$n+1$ n'est pas appariée" $\Rightarrow \forall i \in [1, n-1], \quad a_{n+1\,i} = 0$ alors que "$n+1$ est appariée avec une base i" $\Rightarrow \exists i \in [1, n-1]$ tel que $a_{n+1\,i} = 1$. On en déduit la formule du lemme par addition des contributions des n cas.

Preuve (Théorème 12) : *D'après la récurrence du Lemme 1 :*

$$s_{n+1} = s_n + \sum_{i=1}^{n-2} s_i s_{n-i-1}$$

$$s_{n+1}z^{n+1} = z(s_n z^n) + x^2 \sum_{i=1}^{n-2} s_i s_{n-i-1} z^{n-1}$$

$$\sum_{n\geq 0} s_{n+1}z^{n+1} = z(\sum_{n\geq 0} s_n z^n) + x^2 \sum_{n\geq 0}(\sum_{i=1}^{n-1} s_i s_{n-i-1} z^{n-1} - s_{n-1}s_0 x^{n-1})$$

Alors, si l'on pose $S(z) = \sum_{n\geq 0} s_n z^n$:

$$S(z) - 1 = zS(z) + x^2 S(z)^2 - x^2 S(z)$$

On résoud cette équation et on trouve deux solutions, dont une admet des coefficients positifs dans un développement de Taylor en 0.

$$\Rightarrow S(z) = \frac{z^2 - z + 1 - \sqrt{1 + z(z^3 - 2z^2 - z - 2)}}{2z^2}$$

6.1.3 *Décompositions*

A la recherche d'un algorithme de programmation dynamique pour résoudre le problème du repliement, Waterman propose une décomposition *canonique* des structures secondaires en sous structures.

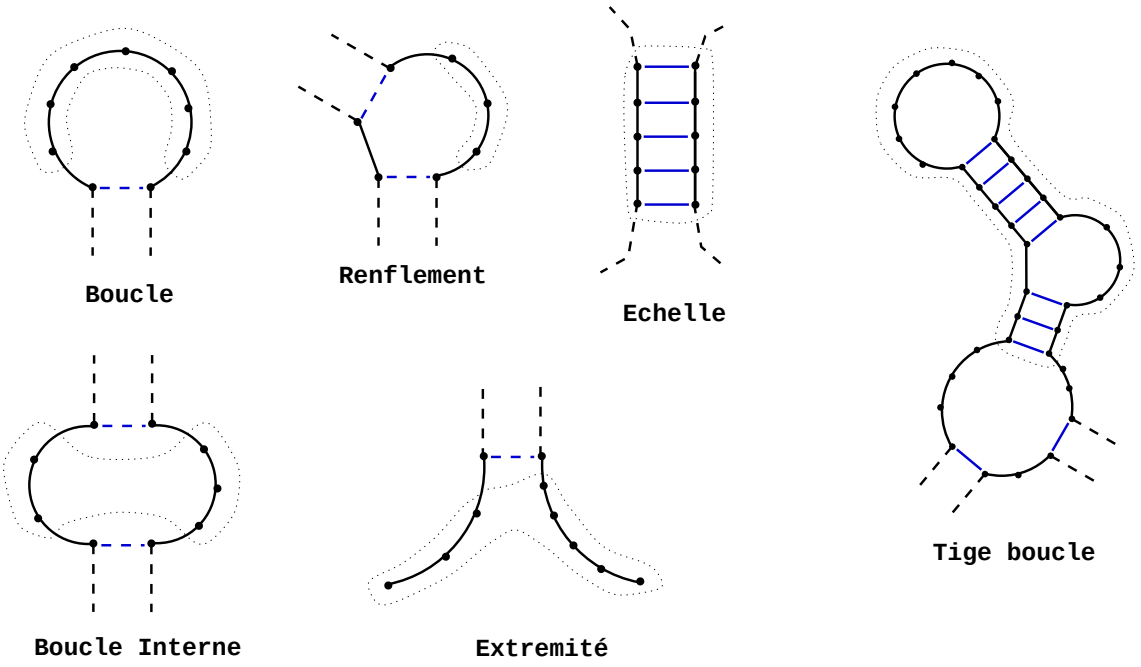


Fig. 6.2 : Les différents type de sous structures
Les bases attribuées aux différentes sous structures sont celles entourées en pointillés.

Définition 19 (Sous-structures d'une structure secondaire) :
Soit $A = (a_{ij})$ *la matrice d'adjacence d'une structure secondaire :*

- **Boucle** *:*
  Formellement, région $R = [i+1, j-1]$ *telle que* $a_{ij} = 1$ *et* $\forall k \in R$, k *est non apparié.*
  Bases non appariées encadrées par une paire (i, j) *de bases appariées.*
  (i, j) *est la fondation de la boucle.*

- **Renflement** *:*
  Région $R = [i+1, j-1]$ *telle que* i *et* j *sont appariés,* $a_{ij} = 0$ *et* $\forall k \in R$, k *est non apparié.*
  Concrètement, bases non appariées séparant deux bases non appariées entre elles.

- **Boucle interne** *:*
  Régions $R = [i+1, j-1]$ *et* $R' = [k+1, l-1]$ *telles que* $a_{il} = 1$, $a_{jk} = 1$ *et* $\forall k \in R \cup R'$, k *est non apparié.*
  Bases séparant deux hélices sur les deux brins.

- **Extremité** *:*
  Région $R = [1, i-1]$ ou $R = [i+1, n]$ telle que i est apparié et $\forall k \in R$, k est non apparié.
  Extrémités pendante de la structure secondaire.

- **Echelle(alt. Hélice)** *:*
  Régions $[i+1, j-1]$ et $[k+1, l-1]$,$j-i = l-k$ telles que $a_{il} = 0$, $a_{jk} = 0$ et $\forall k \in [1, j-i-1], a_{i+k\, l-k} = 1$.
  Paires de base empilées sans renflement.

- **Tige Boucle** *:*
  Plus longue région $[i+1, j+1]$ telle que $a_{ij} = 0$, $a_{i+1\, j-1} = 1$ et contenant exactement une boucle.
  Plus longue séquence encadrée par une paire de bases appariées contenant exactement une boucle.

Remarque : Ces définitions des sous structures diffèrent légèrement de celles proposées par Zucker. Elles portent cependant le même nom, ce qui est une difficulté supplémentaire propre à la bioinformatique, où il n'existe pas encore d'autorité reconnue en matière d'appellation.

Théorème 13 (Décomposition des structures secondaires) :
Toute structure secondaire peut être décomposée de façon unique en :

1. *Boucles, Echelles, Renflements et Extremités.*
2. *Tiges Boucles et Renflements, Extremités et Echelles* **n'appartenant pas à une Tige Boucle**.

Preuve : *On ne prouve que la première décomposition, car ce type de preuve est un peu lourd et redondant.*
Soit A la matrice d'adjacence d'une structure secondaire :
Si i et j sont appariées, alors i appartient indiscutablement à une échelle.
Sinon, soit i une base non appariée et $[k, l] \ni i$, la plus longue séquence de bases non appariées contenant i.
Si $k = 1$ ou $l = n$, alors i appartient à une extremité. En effet, l'appartenance à un renflement ou une boucle impose la présence de bases appariées aux deux extremités, ce qui est ici impossible.
Si $a_{k-1\, l+1} = 1$, alors i appartient à une boucle. Si elle appartenait à un renflement, alors $a_{k-1\, l+1} = 0$.
Sinon, il s'agit bien d'un renflement, car $a_{k-1\, l+1} = 0$.

6.1.4 Ordre d'une structure secondaire

On rappelle que l'étude des structures secondaires d'ARN est motivée historiquement par l'analyse d'algorithmes de repliements par minimisation d'énergie libre. Dans ce contexte, un paramètre apparait comme critique à Waterman : l'ordre. Schématiquement, on peut comparer une structure secondaire à un arbre dont les feuilles sont des boucles, les noeuds sont des boucles multiples et les arêtes sont des échelles. L'ordre d'une structure secondaire est alors la hauteur de cet arbre. La définition de Waterman est beaucoup plus algorithmique :

Définition 20 (Ordre d'une structure secondaire d'ARN) :
Soit A la matrice d'adjacence d'une structure secondaire d'ARN S de taille n.
Soit $(A^{(i)})_{i \geq 0}$ la suite de matrices d'adjacences telles que :

- $A^{(0)} = A$
- $A^{(i+1)}$ *: Décomposer $A^{(i)}$ en Tiges Boucles, Renflements, Extremités et Echelles puis , pour tout $k, l \in [1, n]$:*
  - *Si k et l appartiennent à une tige boucle,s $a_{kl}^{(i)} = 1$ et $k \neq l \pm 1$, alors $a_{kl}^{(i+1)} = 0$.*
  - *Sinon $a_{kl}^{(i+1)} = a_{kl}^{(i)}$.*

L'ordre de S est le plus petit k tel que $A^{(k)}$ est le graphe linéaire, c'est à dire que $\forall i, j \in [1, n], \quad i \neq j \pm 1 \Rightarrow a_{i,j}^{(k)} = 0$

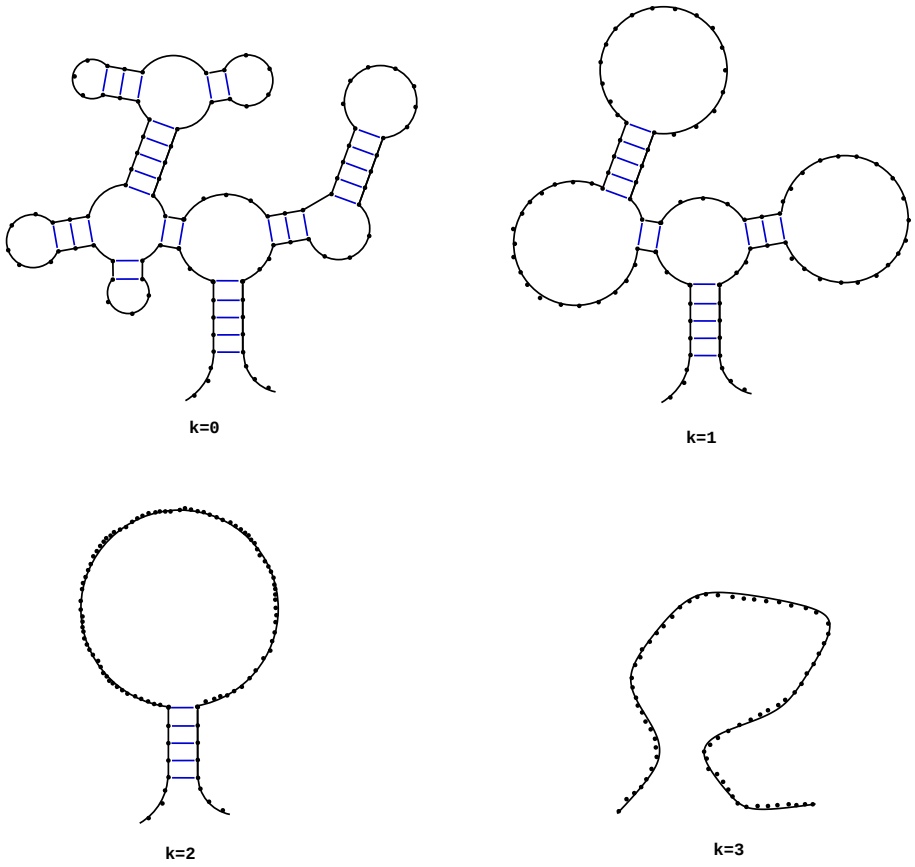


Fig. 6.3 : Extraction de l'ordre d'une structure secondaire

6.2 *Méthodologie DSV pour l'étude de l'ordre[2]*

En 1983, G. Viennot et M. Vauchaussade de Chaumont appliquent la méthodologie DSV et étudient l'ordre des structures secondaires à partir d'une bijection entre les structures secondaires et un sous ensemble des mots de Motzkin.

On rappelle quelques éléments et notations de théorie des langages:

- On appelle **facteur** d'un mot w un mot k tel que $\exists u, v \; tels \; que \; w = u.k.v$
- On note $|w|_k$ le nombres d'occurences de la lettre c dans le mot k.
- w_k est la k-ième lettre de w.

Définition 21 (Mots de Motzkin) :
Les mots de Motkin sont les mots $w \in \{a, b, c\}^*$ tels que :

1. $w = u.v \Rightarrow |u|_a \geq |u|_b$
2. $|w|_a = |w|_b$

6.2.1 *Bijection avec une classe de mots de Motzkin*

Théorème 14 (Bijection Viennot/Vauchaussade de Chaumont) :
Les structures secondaires sont en bijection avec les mots de Motkin sans facteur $a.b$.

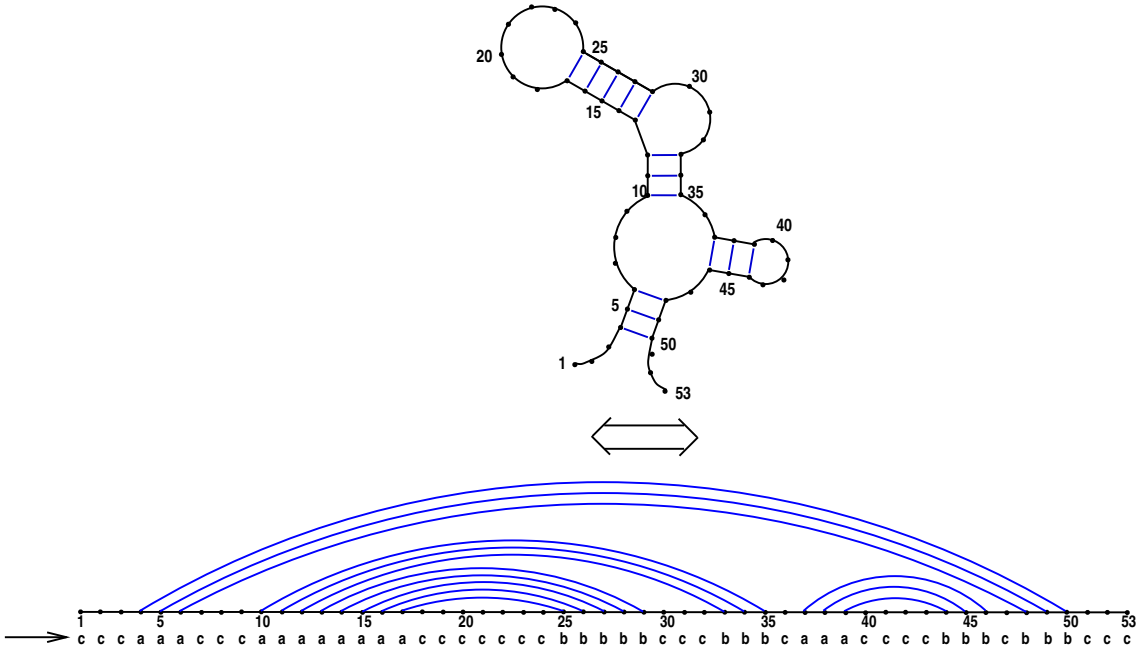


Fig. 6.4 : Transformation structures secondaires/mots de Motzkin

Preuve : *Dans un premier temps, on exhibera un codage des structures secondaires en mots de Motzkin sans facteurs $a.b$. Nous montrerons ensuite que les cardinaux des ensembles de mots et de structures de taille n sont égaux, ce qui prouve la bijectivité du codage.*

Codage : *On code une structure d'ARN S de taille n par un mot w de taille n sur $\{a, b, c\}^*$. Le contenu de la lettre w_k dépend du statut du sommet k :*

- *k non apparié* $\Rightarrow w_k = c$
- *k apparié à $l > k$* $\Rightarrow w_k = a$
- *k apparié à $l < k$* $\Rightarrow w_k = b$

En utilisant la décomposition introduite dans la preuve de la récurrence de Waterman (voir Figure 6.1), on prouve qu'il s'agit bien d'un code, c'est à dire ici d'une fonction injective de l'ensemble des structures secondaires dans l'ensemble des mots de Motzkin sans facteur $a.b$.
On remarque que deux codages de structures secondaires de tailles différentes sont eux mêmes de tailles différentes et donc différents en temps que mots. On va alors comparer les structures de même taille :

- *$n = 1$ et 2 : Il existe deux structures, composées de 1 et 2 bases non appariées à cause du point (iii) de la définition de Waterman. Ces structures sont codées en des mots c et $c.c$, pour $n = 1$ et 2.*
- *Supposons* $S_1 \neq S_2 \Rightarrow w_1 \neq w_2, \forall k = |S_1| = |S_2| \leq n$
- *$|S_1| = |S_2| = k$ et k non apparié :*
  Alors les codages w_1 et w_2 de deux structures secondaires $S_1 \neq S_2$ sont tels que $w_1 = w'_1.c$ et $w_2 = w'_2.c$, avec $w'_1 \neq w'_2$ car sinon $S_1 = S_2$.
- *$|S_1| = |S_2| = k$ et k apparié à $j < k - 1$:*
  Alors les codages w_1 et w_2 de $S_1 \neq S_2$ sont tels que $w_1 = w'_1.a.w''_1.b$ et $w_2 = w'_2.a.w''_2.b$, avec $w'_1 \neq w'_2$ ou $w''_1 \neq w''_2$ car sinon $S_1 = S_2$.

De plus, on peut prouver grâce à cette même récurrence que tout codage w d'une structure secondaire est tel que $w = u.v \Rightarrow |u|_a \geq |u|_b$ et $|w|_a = |w|_b$. Il s'agit donc de mots de Motzkin.
Enfin, k apparié à $j < k - 1$ implique directement la non apparition de motifs $a.b$. On considère les paires de bases appariées les plus proches, c'est à dire celles qui ne délimitent aucun appariement. Seules celles ci sont susceptibles d'être codées en un facteur $a.b$, or celles ci ne peuvent être consécutives, donc il ne peut y avoir de facteur $a.b$ dans les mots du code.

Le codage proposé défini bien une fonction injective. De plus, les mots de Motzkin sont réputés pour être générés par la grammaire non contextuelle suivante :

$$M \ \rightarrow \ a.M.b.M \mid c.M \mid \varepsilon$$

On va d'abord transposer cette règle en relation sur $M = \sum_{w \in L(M)} w$ la série génératrice de $L(M)$ en variables non commutative :

$$M \ = \ aMbM \ + \ cM \ + \ 1$$

Puis on impose l'absence de facteur $a.b$, qui revient trivialement à interdire une réécriture du premier M en ε dans la partie $aMbM$ de la relation. Soit M' la série non commutative des mots de Motzkin sans facteur $a.b$, alors :

$$M' \ = \ a(M' - 1)bM' \ + \ cM' \ + \ 1$$

On applique un morphisme substituant à toutes les lettres la variable z. On résoud alors cette équation du second degrès en M', et, en choisissant la solution de coefficients positifs, on retrouve la série génératrice de dénombremenent des structures secondaires.

$$M' = \frac{z^2 - z + 1 - \sqrt{1 + z(z^3 - 2z^2 - z - 2)}}{2z^2}$$

Par unicité du développement de Taylor, on en déduit qu'il existe une bijection entre les deux familles d'objets étudiés, et que le codage proposé est une bijection.

6.2.2 Apparition de l'ordre dans les mots de Motzkin

Tout d'abord, on transpose le concept de pyramide maximale d'un mot de Dyck aux mots de Motzkin.

Définition 22 (Pyramide maximale d'un mot de Motzkin) :
On appelle pyramide un facteur u d'un mot w de Motzkin tel que:

$$u \in (c^*.a)^k.c^*.(b.c^*)^k$$

u est une pyramide maximale si il ne peut être prolongé en une pyramide $v \ni u$ telle que $|v| > |u|$.

En remarquant, comme le montre la figure 6.5, que les tiges boucles des str. sec. correspondent à des pyramides maximales dans leur codage, on transpose naturellement la notion d'ordre sur les mots de Motzkin. On définit pour cela une fonction Π qui *aplatit* le mot de Motzkin en remplaçant par des c toutes les lettres contenues dans des pyramides maximales. L'ordre est alors le plus petit k tel que $\Pi^k(w) = c^{|w|}$.

On construit alors le système suivant, dans lequel M_k(resp. $M_{\leq k}$) est le non terminal ayant pour langage les mots de Motzkin [1] d'ordre k (resp. $l \leq k$) et U_k(resp. $U_{\leq k}$) le non terminal des mots

[1] sans facteur $a.b \ldots$

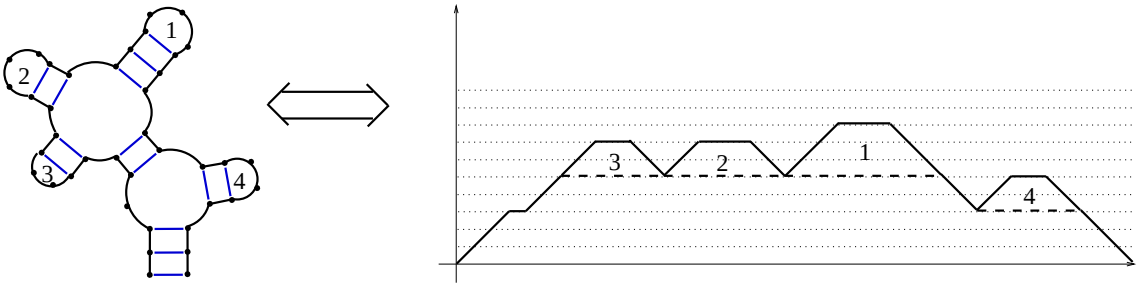


Fig. 6.5 : Les tiges boucles d'une struct. sec. sont en relation avec les pyramides maximales du mot de Motzkin

de Motzkin premiers, c'est à dire non décomposables en séquence de mots de Motzkin, d'ordre k (resp. $l \leq k$).

$$\begin{aligned}
M_k &= M_{\leq k} - M_{\leq k-1} \\
M_0 &= \frac{1}{1-a} \\
M_{\leq k} &= \frac{1}{(1-N_{\leq k})} \\
N_k &= a.M_{\leq k-2}.N_{k-1}.M_{\leq k-2}.N_{k-1}.M_{\leq k-1}.b + a.M_{\leq k-1}.N_k.M_{\leq k-1}.b \\
N_1 &= a.M_0.N_1.M_0.b + a.M_0.N_0.b \\
N_0 &= a
\end{aligned}$$

On résoud ce système en utilisant des propriétés des polynômes orthogonaux et quelques résultats sur les pavages et autres séries génératrices de mots de Dyck de hauteur bornée, et on obtient le résultat suivant :

Théorème 15 (Série génératrice des str. sec. d'ordre *k*) :
La série génératrice des structures secondaires d'ordre k est $M_k(z)$ telle que :

$$M_k(z) = \frac{z^{5.2^{k-1}-2}}{(1-z)\prod_{i=1}^{k} Z_i(z)}$$

Où $Z_{k+1}(z) = Z_k^2 - 2z^{5.2^{k-1}}$ et $Z_1(z) = 1 - 2z - z^3$

Cependant, aucune estimation des coefficients $m_{k,n}$ n'est proposée.

6.3 Application de résultats sur le nombre d'Horton-Strahler[16] à l'étude de l'ordre

6.3.1 Formulation alternative des séries sur l'ordre

Dans l'article de Vauchaussade de Chaumont et Viennot figure aussi les séries génératrices des mots de Dyck d'ordre donné. Les auteurs remarquent que les séries génératrices obtenues sont identiques à celles des mots de Dyck de nombre d'Horton-Strahler donné.
On rappelle sans s'y attarder la définition du nombre d'Horton-Strahler pour un arbre binaire T :

$$hs(T) ::= \begin{cases} 0 & Si\ T\ est\ une\ feuille \\ hs(T_1)+1 & Si\ T=(T_1,T_2)\ et\ hs(T_1)=hs(T_2) \\ \max(hs(T_1),hs(T_2)) & Sinon \end{cases}$$

En reprenant la définition 21, on constate que tout mot obtenu à partir d'un mot de Dyck en y insérant des lettres c est un mot de Motzkin. De plus, si on ajoute un caractère c à chaque occurence du facteur $a.b$, alors on obtient une structure secondaire. Enfin, on remarque que toute les structures secondaires sont engendrables par cette méthode, qui conserve l'ordre [2].
Donc, si on sait compter les mots de Dyck selon leur ordre et leur nombre de facteurs $a.b$ on peut, par changement de variable, compter les structures secondaires selon leur ordre. C'est l'approche que choisit M.Nebel dans [16], après avoir obtenu dans [17] des résultats sur l'analyse des mots de Dyck de nombre d'Orthon-Strahler donné.

Théorème 16 (Séries génératrices alternatives sur l'ordre) :
La série génératrice des structures secondaires non linéaires d'ordre k selon les nombres de bases appariées (resp. non appariées) marquées par un paramètre z (resp. u) est R_k telle que :

$$R_k(z,u) = \frac{\sqrt(u)}{u-1} U_{2^p-1}^{-1} \frac{-(u-1)^2 + (1+u)z^2}{2z^2\sqrt{u}} = \frac{\sqrt{u}(1-\omega)\omega^{2^{p-1}}}{(1-a)\sqrt{\omega}(1-\omega^{2p})}$$

Où

$$\omega := \frac{1-\varepsilon}{1+\varepsilon} \qquad \varepsilon := \sqrt{1 - 4\frac{uz^4}{((a-1)^2-(1+a)z^2)^2}}$$

Et $U_n(z)$ est le n-ième polynôme de Chebyshev de deuxième type, défini par $U(0,x) = 1$, $U(1,x) = 2x$ et $U(n,x) = 2xU(n-1,x) - U(n-2,x)$.

Preuve : *Soit w un mot de Dyck, on lui associe un ensemble de structures secondaires par la transformation β définie récursivement comme suit :*

$$\begin{array}{llll} (1) & a\,b \xrightarrow{\beta} c^*\,a\,c^+\,b\,c^* & (2) & a\,b\,v \xrightarrow{\beta} c^*\,a\,c^+\,b\,\beta(v) \\ (3) & a\,v\,b \xrightarrow{\beta} c^*\,a\,\beta(v)\,b\,c^* & (4) & a\,v\,b\,u \xrightarrow{\beta} c^*\,a\,\beta(v)\,b\,\beta(u) \end{array}$$

Où v et u sont des mots différents de ε. Dans [17], M.Nebel obtient les séries génératrices R_p des mots de Dyck d'un ordre k selon les nombres de sous mots de type (1) (resp. (2)+(3) et (4)) marqués par la variable v(resp. w et x).

$$R_p(x,w,v) = -\frac{v}{\sqrt{xv}} U_{2^p-1}^{-1} \frac{2w-1}{2\sqrt{xv}} = \frac{v(1-\omega)\omega^{2^{p-1}}}{\sqrt{xv}\sqrt{\omega}(1-\omega^{2p})}$$

Avec

$$\omega := \frac{(1 - \sqrt{1 - 4\frac{xv}{(1-2w)^2}})}{(1 + \sqrt{1 - 4\frac{xv}{(1-2w)^2}})}$$

On substitue aux variables les séries génératrices des langages rationnelles images par la transformation β des motifs (1),(2),(3) et (4) .
Les motifs de type (1) étant marqué par la variable v, on effectue la substitution définie par $v := z^2\frac{u}{(1-u)^3}$. Pour une taille et un ordre donné, il existe autant de structures secondaires des formes (2) et (3), car dans ces deux formes, v est un mot de Dyck de taille $n-2$ et d'ordre k. La série génératrice de la transformée de la forme (2) est $z^2\frac{u}{(1-u)^2}$ et celle de la forme (3) $z^2\frac{1}{(1-u)^2}$. On substitue donc $w := \frac{z^2}{2}(\frac{u}{(1-u)^2} + \frac{1}{(1-u)^2})$.
Enfin, la série génératrice de la transformée de (4) est $z^2\frac{1}{(1-u)} := x$.
En simplifiant l'expression obtenue, on obtient bien l'expression du théorème 16. De plus, la structure secondaire linéaire est exclue car la décomposition des mots de Dyck exclut le mot ε.

[2] Intuitivement, parce que la définition de l'ordre ne fait pas intervenir les bases non appariées

6.3.2 Extraction des comportements asymptotiques

A partir de cette formulation utilisant les polynômes de Chebyshev, dont on connait le comportement asymptotique, M.Nebel étudie divers paramètres, dont nous donnerons l'expression quand celle ci est *lisible* :

- **Nombres $a_{n,k}$ de structures secondaires de taille n et d'ordre k :**

$$a_{n,k} \sim z_k^{-n} \frac{-4{z_k}^3}{(1-z_k)(z_k^3-6z_k+5)} \frac{2^{-k}\sin^2(\frac{2^k-1}{2^k}\pi)}{\cos((2^k-1)\pi)}$$

  Où z_k est la plus petite solution réelle de $\frac{z^3+2z-1}{2z^{5/2}} = -\cos(2^{-k}\pi)$.
  De plus, on a : $0 \leq z_k - (\frac{3}{2} - \frac{1}{2}\sqrt{5}) \leq 4^{-k}$

- Nombres $a_{m,n,k}$ des structures secondaires d'ordre k ayant m bases non appariées et n bases appariées.

- Espérance $\mathbb{E}_n(K)$ de l'ordre d'une structure secondaire de taille n :

$$\mathbb{E}_n(K) = \frac{1}{2}\log_2(\frac{2\pi^2}{\rho}n) - \frac{\gamma+2}{2\ln(2)} + \Delta(n) + O(\frac{1}{n})$$

  Où Δ est une fonction oscillante telle que $|\Delta(x)| < 0.0406$ et $\rho := \frac{\sqrt{9\sqrt{5}-20}}{5\sqrt{5}-11}$

- Moments et variance de l'ordre d'une structure secondaire aléatoire.

- Espérance $\mathbb{E}_n(M)$ et variance $\mathbb{V}_n(M)$ du nombre de bases non appariées dans une str. sec. de taille n :

$$\mathbb{E}_n(M) \sim \frac{n}{\sqrt{5}} + \frac{3}{10} + \frac{1}{\sqrt{5}} + O(\frac{1}{n})$$

$$\mathbb{V}_n(M) \sim \frac{2}{5} + \frac{3}{10}\frac{n+1}{\sqrt{5}} + \frac{1}{100}$$

- Espérance $\mathbb{E}_{n,k}(M)$ du nombre de bases non appariées dans une str. sec. de taille n et d'ordre k :

$$\mathbb{E}_{n,k}(M) \sim n(1 + \frac{4-4z_p}{z_p^2+z_p-5})$$

- Espérance $\mathbb{E}_n(T)$ du nombre de tiges boucles dans une str. sec. de taille n :

$$\mathbb{E}_n(T) \sim (1 - \frac{2}{5}\sqrt{5})n + \frac{13}{20} - \frac{3}{20}\sqrt{5} + O(\frac{1}{n})$$

- Espérance $\mathbb{E}_{n,k}(T)$ du nombre de tiges boucles dans une str. sec. de taille n et d'ordre k :

$$\mathbb{E}_{n,k}(T) \sim n(1 + \frac{4}{z_p^2+z_p-5})$$

- Espérance $\mathbb{E}_n(NT)$ du nombre de bases non appariées localisées dans une tiges boucles dans une str. sec. de taille n :

$$\mathbb{E}_n(NT) \sim \frac{1}{2} + \frac{1}{2}\sqrt{5} + \frac{1}{2n} + O(\frac{1}{n^2})$$

- Espérance $\mathbb{E}_{n,k}(NT)$ du nombre de bases non appariées localisées dans une tiges boucles dans une str. sec. de taille n et d'ordre k :

$$\mathbb{E}_{n,k}(NT) \sim \frac{1}{1-z_k} \underset{k\to\infty}{\sim} \frac{1}{2} + \frac{1}{2}\sqrt{5}$$

- Espérance $\mathbb{E}_n(R)$ du nombre de renflements dans une str. sec. de taille n :

$$\mathbb{E}_n(R) \sim \frac{3\sqrt{5}-5}{10}n - \frac{61}{20} + \frac{21}{20}\sqrt{5} + O(\frac{1}{n})$$

- Espérance $\mathbb{E}_{n,k}(R)$ du nombre de renflements dans une str. sec. de taille n et d'ordre k :

$$\mathbb{E}_n(R) \sim n\frac{z_k(3+z_k(z_k-3)))}{5-z_k-z_k^2}$$

- Espérance $\mathbb{E}_n(NR)$ du nombre de bases non appariées localisées dans un renflement d'une str. sec. de taille n :

$$\mathbb{E}_n(NR) \sim \frac{1}{2} + \frac{1}{2}\sqrt{5} + \frac{1035 + 278\sqrt{5} - \sqrt{1301230 + 532980\sqrt{5}}}{160n}$$

- Espérance $\mathbb{E}_{n,k}(NR)$ du nombre de bases non appariées localisées dans un renflement d'une str. sec. de taille n et d'ordre k :

$$\mathbb{E}_{n,k}(NR) \sim \frac{1}{1-z_k} \underset{k\to\infty}{\sim} \frac{1}{2} + \frac{1}{2}\sqrt{5}$$

7. CONTRIBUTION

Les formalismes ayant permis l'études des structures secondaires d'un point de vue combinatoire sont assez peu adaptés à la génération aléatoire. L'ensemble des graphes utilisés par Waterman dans [24] est un sous ensemble des graphe connexes non orientés, pour laquelle la génération aléatoire uniforme est déjà un problème ardu, à cause des problèmes d'homomorphismes. La modélisation par grammaire de Vauchaussade de Chaumont et Viennot dans [2] pose quand à elle deux problèmes. D'abord, la grammaire proposée utilise une soustraction, ce qui la fait sortir du domaine non contextuel[1]. Ensuite, elle n'est pas satisfaisante car il est difficile d'y poser des *marqueurs* permettant la distinction des différentes sous structures. La démarche de Nebel n'inclut quand à elle pas vraiment d'étape de modélisation, mais plutôt une succession d'astuces combinatoires et de techniques d'analyses asymptotiques.

On choisit donc de proposer un nouveau modèle pour les structures secondaires d'ARN, basé sur une grammaire non contextuelle. Ce formalisme parait être le meilleur outil pour ce que nous nous proposons d'étudier. En effet, toutes les séries génératrices manipulées jusqu'ici dans l'étude des structures secondaires sont algébriques, donc les grammaires non contextuelles sont un formalisme suffisemment expressif. De plus, il existe une grande quantité de littérature traitant de la génération ayant pour origine ces objets. Enfin, les principes de calculs asymptotiques automatiques décrits en [9], s'appliquant entre autres aux séries génératrices des langages non contextuels, ont été implémentés dans une bibliothèque de fonctions Maple développée au sein de l'INRIA Rocquencourt. On pourra donc, sans trop s'abrutir de calculs, retrouver la plupart des résultats de Waterman et Nebel, et étendre en rajoutant des paramètres dans le modèle comme le nombre minimal de bases dans une boucle ou bien la taille minimale d'une échelle.

7.1 Modélisation des structures secondaire par une grammaire non contextuelle

7.1.1 Adaptation de la décomposition de Waterman

Cette nouvelle décomposition s'appuie sur une interprétation arborescente des structures secondaires.
On rappelle le principe de la première décomposition de Waterman :

Toute structure secondaire peut être décomposée de façon unique en Boucles, Echelles, Renflements et Extremités.

On va introduire de nouveaux paramètres dans cette décomposition de Waterman :

- **Les Bourgeons** :
  Il s'agit d'un renflement séparant deux échelles sur un seul des deux brins. Formellement, $[i, j]$ tel que $i - 1$ est apparié à k et $j + 1$ est apparié à l tel que $k = l + 1$ et $\forall m \in [i, j], m$ est non apparié.

[1] Cependant on peut facilement construire une grammaire non contextuelle qui reconnaisse le même langage.

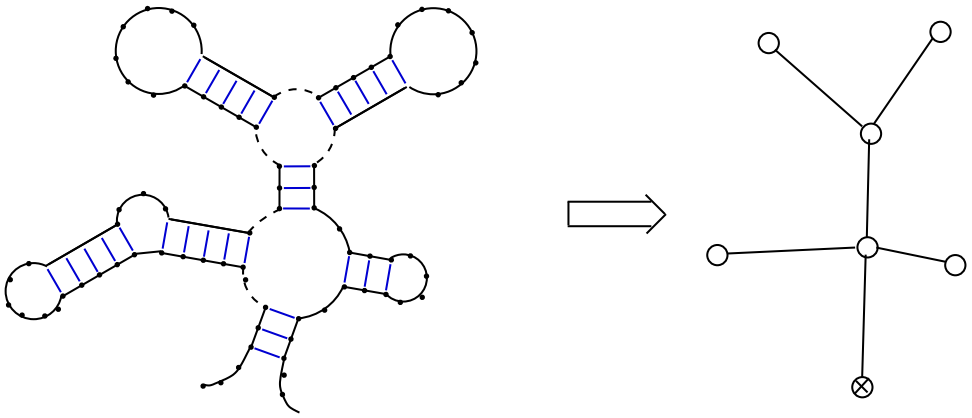


Fig. 7.1 : Interprétation arboresente des structures secondaire

- **Les Boucles Internes** :
  Dans une approche *arborescente* des structures secondaires (voir Figure 7.1), il s'agit des noeuds ayant un facteur de branchement égal à 1. Si un renflement de la décomposition initiale peut être considéré comme une boucle interne au sens de la définition de Waterman, alors il appartient, dans notre nouvelle décomposition, à une boucle interne.
  $[i, j]$ et $[k, l]$ tels que $i - 1$ apparié à $l + 1$, $j + 1$ apparié à $k - 1$ et, $\forall m \in [i, j] \bigcup [k, l]$, m est non apparié.

- **Les Multiboucles** :
  Il s'agit des noeuds de degrès de branchement supérieur à 2. Appartiennent à une multiboucle les bases qui séparent les échelles issues de ce noeud.
  Dans une approche plus orientée graphes, régions $A_1 = [i_1, j_1]$, $A_2 = [i_2, j_2]$,. . .,$A_k = [i_k, j_k]$, $k \geq 3$ et $i_1 < j_1 < i_2 < j_2 < \ldots < i_k < j_k$ telles que $\forall l \in [1, k - 1]$, $a_{j_l+1, i_{l+1}-1} = 1$, $a_{i_1-1, j_k+1} = 1$ et $\forall l \in [1, k], \forall m \in [i_l, j_l]$, m est non apparié.
  On appellera **remplissage** chaque région A_i prise individuellement.

Théorème 17 (Nouvelle décomposition) :
Toute structure secondaire peut être décomposée uniquement en Boucles, Echelles, Extrémités, Boucles Internes, Multiboucles et Bourgeons.

Preuve : *On remarque d'abord que les nouvaux éléments introduits dans la définition sont tous à l'origine des Renflements, or on sait qu'une décomposition unique en Renflements, Boucles, Extrémités et Echelles existe. Pour prouver l'existence et l'unicité de la décomposition, il suffit donc de prouver que tout renflement dans la décomposition de Waterman devient sans ambiguité une des trois nouvaux types de sous structures.*

Tout renflement $[i + 1, j - 1]$ est délimité par deux bases $i < j$ appariées à des bases l, k telles que $i < j < k < l$ par la propriété iii) de la définition de Waterman :

- *Soit $k = l - 1$, alors $[i + 1, j - 1]$ est un Bourgeon.*
- *Sinon Si les bases de $[k + 1, j - 1]$ sont non appariées, alors $[i + 1, j - 1] \cup [k + 1, j - 1]$ constitue une Boucle Interne.*
- *Sinon on prouve par construction qu'il existe une décomposition unique de $[i + 1, l - 1]$ telle que :*

$$[i + 1, l - 1] = [i + 1, X_1 - 1] \cup [X_1, X_2] \cup \ldots \cup [X_k + 1, l - 1], \quad k \geq 4, k \equiv 0[2]$$

Où $[i+1, X_1 - 1], [X_{2j}+1, X_{2j+1}-1], [X_k+1, l-1]$ *sont des bases non appariées et* X_{2j-1} *apparié avec* X_{2j}.
Pour cela, on parcourt la structure primaire de $i+1$ *vers* $j-1$.

- *Tant qu'on ne rencontre que des bases non appariées, on les concatène à un ensemble de base non appariées(*$[X_j+1, X_{j+1}-1]$*)*
- *Quand on rencontre une base appariée* $X_p \neq l$*, on saute vers la base* X_{p+1} *à laquelle elle est directement appariée.*

Par induction, on prouve que $X_{p+1} > X_p$*, car sinon il existe* $h < p$ *tel que* $X_h < X_p < X_{h+1} < X_{p+1}$*, ce qui viole la condition iii) de la définition de Waterman. De plus, on atteint bien* $l-1$ *car si on dépasse* $l-1$*, alors il existe* r *tel que* $X_r < l-1 < X_{r+1}$*. Or, pour tout* r*,* $j+1 < X_r < l-1 < X_{r+1}$*, ce qui viole de nouveau la condition iii) de Waterman.*

Donc cette décomposition de la région $[i+1, l-1]$ *existe bien. Elle est unique car s'il en existe une deuxième, alors considèrons le premier* X_j *dans une lecture de gauche à droite pour lequel les décompositions diffèrent. Soient* $X_j^1 < X_j^2$ *les deux valeurs distinctes pour* X_j*, alors on aboutit à une contradiction car si* X_j^2 *est l'extrémité d'une séquence de bases non appariée, alors* X_j^1 *est non apparié, ce qui est absurde, et si* X_j^1 *est l'origine d'une séquence de bases non appariées, alors* X_j^2 *est non apparié. Donc* $X_j^1 = X_j^2$ *et la décomposition est unique.*
Or la décomposition implique directement l'appartenance à une multiboucle.

On veut *marquer* les différents types de sous structures dans la structure secondaire. On va donc *coller* à cette décomposition inspirée de Waterman en utilisant des symboles terminaux différents pour les bases appartenant aux différentes sous structures.

7.1.2 Grammaire non contextuelle et problème d'ambiguité structurelle

On devra faire attention à ne pas introduire d'ambiguité dans la grammaire. En effet, même si les grammaires proposées ne sont pas ambiguës au sens de la théorie des langages, l'utilisation de vocabulaires distincts pour les différentes sous structures est susceptible de projeter deux séquences différentes sur la même structure. Pour éviter cela, on va interdire certaines successions.
Dans la suite, on dira qu'une sous structure b succède à une sous structure a si elles sont adjacentes et si un parcours en profondeur de l'arborescence de la structure secondaire rencontre a avant b.

Par exemple, si une échelle peut succéder à une échelle, alors on introduit de l'ambiguité dans la grammaire, car la structure contenant une échelle de n bases peut être engendrée par la séquence constituée d'une échelle de n base, ou bien par la séquence constituée de deux échelles de tailles respectives n et $n-k$, etc ... D'autre part, cette ambiguité ne peut être compensée par un facteur multiplicateur, car la structure Boucle sur n bases n'est elle engendrée que par la séquence de n caractères non appariés.
On présente donc dans le tableau 7.2 les admissibilités et les interdictions des successions des sous structures. On propose alors la grammaire de la Figure 7.3, de symbole initial `ARN`, qui respecte ces contraintes.

7.1.3 Validation de la grammaire proposée

On va prouver que la grammaire engendre bien, à un renommage des lettres près, les structures secondaires d'ARN.

Succède ↱	**B**	**E**	**R**	**Ex**	**BI**	**MB**
Boucle(B)	N	N	N	N	N	N
Echelle(E)	O	N	O	N	O	O
Renflement(R)	N	O	N	N	N	N
Extremité(Ext)	N	O	N	N	N	N
Boucle Interne(BI)	N	O	N	N	N	N
MultiBoucle(MB)	N	O	N	N	N	N

Fig. 7.2 : L'admissibilité des adjacences de sous structures

```
      ARN  ->  Ex5 E Ex3
            |  B
            |  Ex5 E MB E SuiteMB Ext3
            |  ε
      Ex3  ->  t3 Ex3 | ε
      Ex5  ->  t5 Ex5 | ε
        E  ->  a E2 b
       E2  ->  a E2 b
            |  R E
            |  E R
            |  BI E BI
            |  B
            |  MB E MB E SuiteMB MB
SuiteMB  ->  MB E SuiteMB | ε
        R  ->  c R | c
       BI  ->  d BI | d
        B  ->  e B | e
       MB  ->  f MB | ε
```

Fig. 7.3 : Une grammaire non contextuelle basée sur la décomposition en sous structures.

Théorème 18 (Validité de la grammaire) :
Soit φ le morphisme défini de $\{a, b, c, d, e, f\}$ dans $\{a, b, c\}$ par :

$$\begin{aligned} \varphi(a) &= a \\ \varphi(b) &= b \\ \varphi(x) &= c, \quad \forall x \in \{c, d, e, f\} \end{aligned}$$

et étendu de $\{a, b, c, d, e, f\}^*$ dans $\{a, b, c\}^*$ par :

$$\varphi(x.\omega) = \varphi(x).\varphi(\omega), \quad \forall x \in \{a, b, c, d, e, f\}$$

La restriction de φ à l'ensemble $\mathcal{ARN}$ des mots engendrés par la grammaire est un isomorphisme de $\mathcal{ARN}$ dans l'ensemble des mots de Motzkin sans facteur $a.b$, en bijection avec les structures secondaires.

Preuve : *Afin de valider cette grammaire, on va procéder en deux étape : On va d'abord prouver que les mots engendrés par la grammaire sont envoyés par le morphisme φ sur des mots de Motzkin sans facteur $a.b$, ce qui implique une relation de cardinalité sur les ensembles . On montre ensuite l'égalité des séries génératrices ordinaires, ce qui prouve le fait qu φ est bien un isomorphisme.*

Soit $\omega' = \varphi(\omega)$ *:*

- $\omega' \in \{a, b, c\}^*$ *:*
  Trivialement, φ *envoie toutes les lettres de* $\{c, d, e, f\}$ *sur* $\{c\}$*, donc le mot obtenu est bien sur* $\{a, b, c\}^*$*.*

- $|\omega'|_a = |\omega'|_b$ *:*
  Toute réécriture d'un non terminal faisant apparaître un a *fait apparaître un* b*, donc on prouve cette propriété par récurrence sur le nombre de réécritures d'un non terminal nécessaire pour engendrer un mot.*

- *Pour toute factorisation* $\omega' = u.v$*,* $|u|_a \geq |u|_b$ *:*
  Toutes les règles engendrant un b *engendre d'abord un* a *lui correspondant. On pourra prouver cette propriété pour les deux règles faisant apparaître* a *et* b *par récurrence sur la longueur des mots engendrés par des non terminaux.*

- ω' *ne contient pas de facteur* $a.b$ *:*
  Les occurences de a *et* b *dans la grammaire sont systématiquement séparées par un non terminal* E_2 *.*
  Or, aucune des règles réécrivant E_2 *ne dérive le mot* ε*.*

$\Rightarrow$ ω' *est bien un mot de Motzkin sans facteur* $a.b$*.*

De plus, on peut obtenir la série génératrice de dénombrement de $\mathcal{ARN}$*, c'est à dire la série comptant les*

mots selon leurs tailles en utilisant les relations induites par les règles de la grammaire[2].

$$
\begin{aligned}
Ex3(z) &= zEx3(z)+1 = \frac{1}{1-z}\\
Ex5(z) &= zEx5(z)+1 = \frac{1}{1-z}\\
R(z) &= zR(z)+z = \frac{z}{1-z}\\
BI(z) &= zBI(z)+z = \frac{z}{1-z}\\
B(z) &= zB(z)+z = \frac{z}{1-z}\\
MB(z) &= zMB(z)+1 = \frac{1}{1-z}\\
SuiteMB(z) &= MB(z)E(z)SuiteMB(z)+1 = \frac{z-1}{1-z-E(z)}\\
E(z) &= zE_2(z)z\\
E_2(z) &= zE_2(z)z\\
&+ R(z)E(z)+E(z)R(z)\\
&+ BI(z)E(z)BI(z)\\
&+ B(z)\\
&+ MB(z)E(z)MB(z)E(z)SuiteMB(z)MB(z)\\
&= \frac{-z^2-z+1-\sqrt{z^4-2\,z^3-z^2-2\,z+1}}{2\,z^2}\\
ARN(z) &= Ex5(z)E(z)Ex3(z)+B(z)\\
&+ Ex5(z)E(z)MB(z)E(z)SuiteMB(z)Ext3(z)+1\\
&= \frac{2}{1-z+z^2+\sqrt{z^4-2\,z^3-z^2-2\,z+1}}
\end{aligned}
$$

Remarques : *$ARN(z)$ est une autre écriture de la série génératrice des structures secondaires, on l'obtient en appliquant une identité remarquable $(a+b)(a-b)=a^2+b^2$ au numérateur et au dénominateur. D'autre part, $ARN(z)-E_2(z)=1$, ce qui s'explique par le fait qu'une paire $a.b$ peut être l'origine de n'importe quelle structure secondaire, sauf une boucle de taille 0.*

En résumé, il existe un morphisme φ envoyant les mots $\mathcal{ARN}$ engendrés par la grammaire dans un sous ensemble des structures secondaires $\mathcal{M}$ ($\Rightarrow |\mathcal{ARN}| < |\mathcal{M}|$). D'autre part, l'égalité des séries génératrices implique l'égalité des cardinaux des sous ensembles pour une taille donnée. on en déduit l'égalité des cardinaux, car une inégalité des cardinaux implique l'existence dans $\mathcal{M}$ d'un mot de taille n qui n'admettrait pas d'antécedant par φ dans $\mathcal{ARN}$, ce qui est impossible car les cardinaux pour une taille n sont égaux. Donc φ est bien un isomorphisme.

[2] Normalement, on doit d'abord s'assurer que la grammaire est non ambiguë avant d'en déduire des relations sur les séries génératrices. Ici, on fait l'inverse : On construit une série génératrice comptant les cardinaux d'éventuels multiensembles (ambiguité), et on compare la série solution avec celle issue d'une grammaire notoirement non ambiguë pour conclure sur la non ambiguité.

7.2 Comportements asymptotiques des distributions des symboles

7.2.1 Méthode

Dans le chapitre précédent, on a résolu le système correspondant à la série génératrice de dénombrement des structures secondaires. Pour cela, on a remplacé toutes les variables correspondant aux divers symboles terminaux par la variables z. On peut aussi décider de conserver les différentes variables comme des marqueurs, ce qui permet une étude des distributions des symboles terminaux.

Théorème 19 (Esperance du nombre d'occurences d'une lettre) :
Soit $x_1, \dots, x_k$ les variables associées aux symboles terminaux $s_1, \dots, s_k$ dans la série génératrice $L(z, x_1, \dots, x_k)$ d'un langage $\mathcal{L}$.
Alors l'espérance $\mathbb{E}_n(N_{s_i})$ du nombre de symboles s_i dans un mot de $\mathcal{L}$ de longeur n pris uniformément est donnée par :

$$\mathbb{E}_n(N_{s_i}) = \frac{[z^n]\frac{\partial L(z,x_1,\dots,x_k)}{\partial x_i}(z,1,\dots,1)}{[z^n]L(z,1,\dots,1)}$$

Preuve : *Par définition, l'espérance d'une variable aléatoire $\mathbb{E}_n(N_i)$ est la valeur moyenne de cette variable sur tous les tirages possibles.*
Plus formellement :

$$\mathbb{E}_n(N_{s_i}) = \frac{\sum_{\omega\in\mathcal{L}_n} |\omega|_{s_i}}{|\mathcal{L}_n|}$$

Or, en passant à la série commutative du langage $\mathcal{L}$, on trouve :

$$L(z, x_1, \dots, x_k) = \sum_{\omega\in\mathcal{L}} z^{|\omega|} x_1^{|\omega|_{s_1}} \dots x_k^{|\omega|_{s_k}}$$

$$\frac{\partial L(z, x_1, \dots, x_k)}{\partial x_i} = \sum_{\omega\in\mathcal{L}} z^{|\omega|} x_1^{|\omega|_{s_1}} \dots |\omega|_{s_i} x_i^{|\omega|_{s_i}-1} \dots x_k^{|\omega|_{s_k}}$$

$$\frac{\partial L(z, x_1, \dots, x_k)}{\partial x_i}(z, 1, \dots, 1) = \sum_{\omega\in\mathcal{L}} z^{|\omega|} |\omega|_{s_i}$$

$$[z^n]\frac{\partial L(z, x_1, \dots, x_k)}{\partial x_i}(z, 1, \dots, 1) = \sum_{\omega\in\mathcal{L}_n} |\omega|_{s_i}$$

On retrouve alors l'expression à prouver en remarquant que :

$$L(z, 1, \dots, 1) = \sum_{\omega\in\mathcal{L}} z^{|\omega|} = \sum_{n\geq 0} |\mathcal{L}_n| z^n$$

On a prouvé la définition alternative de l'espérance du théorème 19.

De plus, on rappelle que, soit $f_1(z)$ et $f_2(z)$ telles que :

$$[z^n]f_1(z) \sim \frac{\rho^n k_1}{n^\alpha} + \frac{\rho^n k_2}{n^{\alpha-1}} + O(\frac{\rho^n}{n^{\alpha-2}})$$

$$[z^n]f_2(z) \sim \frac{\rho^n l_1}{n^\beta} + \frac{\rho^n l_2}{n^{\beta-1}} + O(\frac{\rho^n}{n^{\beta-2}})$$

Alors on a, pour $0 < \alpha < \beta$:

$$\frac{[z^n]f_1(z)}{[z^n]f_2(z)} \sim \frac{k_1}{l_1} n^{\beta-\alpha} + \frac{k_2 l_1 - k_1 l_2}{l_1^2} n^{\beta-\alpha-1} + O(n^{\beta-\alpha-2})$$

On utilise donc la méthode suivante pour caractériser le comportement asymptotique de l'espérance d'un symbole non terminal s_i :

- On calcule la série génératrice $L(z, x_1, \ldots, x_k)$ du langage engendré par la grammaire.
- On calcule l'asymptotique du nombre de structures secondaires de taille n à partir de la série de dénombrement obtenue par Waterman et al. :

$$\Rightarrow [z^n]S(z) \sim \frac{\rho^n l_1}{n^{3/2}} + \frac{\rho^n l_2}{n^{5/2}} + O(\frac{\rho^n}{n^{7/2}})$$

  Avec $\rho = \frac{\sqrt{5}+3}{2}$
- On calcule l'asymptotique du nombre de symboles s_i dans toutes les structures secondaires de taille n :

$$\Rightarrow [z^n]\frac{\partial L(z, x_1, \ldots, x_k)}{\partial x_i}(z, 1, \ldots, 1) \sim \frac{\rho^n k_1}{n^{1/2}} + \frac{\rho^n k_2}{n^{3/2}} + O(\frac{\rho^n}{n^{5/2}})$$

  Remarque : Pour $s_i = t_3$ et $s_i = t_5$, on a $k_1 = 0$
- On en déduit l'asymptotique de l'espérance $\mathbb{E}_n(N_{s_i})$ du nombre de symboles s_i dans une structure secondaire de taille n :

$$\Rightarrow \mathbb{E}_n(N_{s_i}) = \frac{[z^n]\frac{\partial L(z, x_1, \ldots, x_k)}{\partial x_i}(z, 1, \ldots, 1)}{[z^n]S(z)} \sim \frac{k_1}{l_1}n + \frac{k_2 l_1 - k_1 l_2}{l_1^2} + O(\frac{1}{n})$$

7.2.2 *Résultats*

On obtient les résultats résumés dans la figure 7.4. On remarquera que la somme de ces quantités est bien égale à $n + O(\frac{1}{n})$ ce qui est assez rassurant ...

Sous Structure	s	$\mathbb{E}_n(N_s)$
Extrémité 5'	$t5$	$\frac{\sqrt{5}-1}{2} + O(\frac{1}{n})$
Extrémité 3'	$t3$	$\frac{\sqrt{5}-1}{2} + O(\frac{1}{n})$
Echelle brin 5'	a	$\frac{(5-\sqrt{5})n}{10} - \frac{3+2\sqrt{5}}{20} + O(\frac{1}{n})$
Echelle brin 3'	b	$\frac{(5-\sqrt{5})n}{10} - \frac{3+2\sqrt{5}}{20} + O(\frac{1}{n})$
Renflement	c	$\frac{(25-11\sqrt{5})n}{5} + \frac{\sqrt{5}-1}{20} + O(\frac{1}{n})$
Boucle Interne	d	$\frac{2(9\sqrt{5}-20)n}{5} + \frac{53-23\sqrt{5}}{20} + O(\frac{1}{n})$
Boucle	e	$\frac{(3\sqrt{5}-5)n}{10} + \frac{\sqrt{5}+9}{20} + O(\frac{1}{n})$
MultiBoucle	f	$\frac{(7-3\sqrt{5})n}{2} - \frac{7-\sqrt{5}}{4} + O(\frac{1}{n})$

Fig. 7.4 : Distribution asymptotique des bases dans les sous structures d'une structure secondaire.

7.3 Marquage des différentes sous structures

Comme on peut le vérifier, chaque type de sous structure est engendrée de façon canonique par la réécriture d'un non terminal. On peut donc *marquer* les occurences de sous structures avec des variables spécifiques permettant alors l'étude des nombres d'occurences de sous structures à l'asymptotique.

Remarque : Les symboles introduits dans la grammaires sont des mots de taille 0, on ne les marquera pas avec une variable z quand on transposera les règles de réécriture en équations sur les séries génératrices des non terminaux.

On obtient alors la grammaire de la Figure 7.5, qui permet de déduire les résultats de la figure 7.4 par application de la méthode présentée précédemment. On rappelle que les remplissages

$$
\begin{array}{rcl}
\texttt{ARN} & \texttt{->} & \texttt{Ex5}\ m_E\ \texttt{E Ex3} \\
 & | & m_B\ \texttt{B} \\
 & | & \texttt{Ex5}\ m_E\ \texttt{E}\ m_M M\ m_{MB}\ \texttt{MB}\ m_E\ \texttt{E SuiteMB Ext3} \\
 & | & \varepsilon \\
\texttt{Ex3} & \texttt{->} & \texttt{t3 Ex3} \mid \varepsilon \\
\texttt{Ex5} & \texttt{->} & \texttt{t5 Ex5} \mid \varepsilon \\
\texttt{E} & \texttt{->} & \texttt{a}\ \mathtt{E}_2\ \texttt{b} \\
\mathtt{E}_2 & \texttt{->} & \texttt{a}\ \mathtt{E}_2\ \texttt{b} \\
 & | & m_R\ \texttt{R}\ m_E\ \texttt{E} \\
 & | & m_E\ \texttt{E}\ m_R\ \texttt{R} \\
 & | & m_{BI}\ \texttt{BI}\ m_E\ \texttt{E BI} \\
 & | & m_B\ \texttt{B} \\
 & | & m_{MM}\ m_{MB}\ \texttt{MB}\ m_E\ \texttt{E}\ m_{MB}\ \texttt{MB}\ m_E\ \texttt{E SuiteMB}\ m_{MB}\ \texttt{MB} \\
\texttt{SuiteMB} & \texttt{->} & m_{MB}\ \texttt{MB}\ m_E\ \texttt{E SuiteMB} \mid \varepsilon \\
\texttt{R} & \texttt{->} & \texttt{c R} \mid \texttt{c} \\
\texttt{BI} & \texttt{->} & \texttt{d BI} \mid \texttt{d} \\
\texttt{B} & \texttt{->} & \texttt{e B} \mid \texttt{e} \\
\texttt{MB} & \texttt{->} & \texttt{f MB} \mid \varepsilon
\end{array}
$$

Fig. 7.5 : Marquage des sous structures

des multiboucles sont les séquences de bases non appariées séparant les échelles partant d'une multiboucle. On déduit donc du nombre moyen de remplissages le degrès moyen de l'arbre associé à une structure secondaire.
En outre, on retrouve des résultat de Nebel sur le nombre moyen de tiges boucles dans une structure secondaire, en remarquant que chaque tige boucle contient exactement une boucle et chaque boucle est contenu dans exactement une tige boucle. On retrouve aussi le nombre moyen de bases non appariées :

$$\mathbb{E}_n(N_c) + \mathbb{E}_n(N_d) + \mathbb{E}_n(N_e) + \mathbb{E}_n(N_f) \sim \frac{n}{\sqrt{5}} + \frac{3}{10} + \frac{1}{\sqrt{5}} + O(\frac{1}{n})$$

7.4 Contraintes supplémentaires

On peut aussi suivre une démarche similaire à celle de M Régnier dans [19] et vouloir étudier les comportements asymptotiques des différents paramètres en contraignant les tailles minimales

Sous Structure	Nombre	Taille
Echelle	$\frac{5-\sqrt{5}}{5}n - \frac{2\sqrt{5}+3}{10} + O(\frac{1}{n})$	$2 + O(\frac{1}{n})$
Boucle	$\frac{5-2\sqrt{5}}{5}n - \frac{3\sqrt{5}-13}{20} + O(\frac{1}{n})$	$\frac{1+\sqrt{5}}{2} + O(\frac{1}{n})$
Renflement	$\frac{18\sqrt{5}-40}{5}n + \frac{57\sqrt{5}-127}{20} + O(\frac{1}{n})$	$\frac{1+\sqrt{5}}{2} + O(\frac{1}{n})$
Boucle Interne	$\frac{65-29\sqrt{5}}{10}n + \frac{63-28\sqrt{5}}{20} + O(\frac{1}{n})$	$1 + \sqrt{5} + O(\frac{1}{n})$
MultiBoucle	$\frac{7\sqrt{5}-15}{10}n + \frac{129-69\sqrt{5}}{160} + O(\frac{1}{n})$	$\sqrt{5} + O(\frac{1}{n})$
Remplissage MultiBoucle	$\frac{5+\sqrt{5}}{2} + O(\frac{1}{n})$	$\frac{\sqrt{5}-1}{2} + O(\frac{1}{n})$

Fig. 7.6 : Nombres de sous structures et taille d'une occurence de sous structure dans une structure secondaire de taille n. Le *nombre* des remplissages est le nombre moyen de remplissages par MultiBoucle.

e et b des échelles et des boucles. Une telle contrainte s'injecte aisément dans la grammaire, dont on modifie les règles :

$$\texttt{E} \;\texttt{->}\; \mathbf{a^e}\ \texttt{E}_2\ \mathbf{b^e}$$
$$\texttt{R} \;\texttt{->}\; \texttt{c R} \mid \mathbf{c^b}$$

Plusieurs statistiques des sous structures pour différentes valeurs de e et b sont proposées en annexe. On ne résiste cependant pas à présenter le cas $e = 1$ et $b = 0$, qui se ramène aux mots de Motzkin, et pour qui les proportions de sous structures sont des nombres rationnels.

Sous Structure	Taille Totale	Nombre	Taille Unitaire
Extrémité 5'	$\frac{1}{2} + O(\frac{1}{n})$	1	$\frac{1}{2} + O(\frac{1}{n})$
Extrémité 3'	$\frac{1}{2} + O(\frac{1}{n})$	1	$\frac{1}{2} + O(\frac{1}{n})$
Echelle	$(\sqrt{5}-2)n - \frac{9\sqrt{5}-23}{8} + O(\frac{1}{n})$	$\frac{5+3\sqrt{5}}{5} + O(\frac{1}{n})$	
Boucle	$\frac{5-2\sqrt{5}}{5}n - \frac{3\sqrt{5}-13}{20} + O(\frac{1}{n})$	$\frac{1+\sqrt{5}}{2} + O(\frac{1}{n})$	
Renflement	$\frac{18\sqrt{5}-40}{5}n + \frac{57\sqrt{5}-127}{20} + O(\frac{1}{n})$	$\frac{1+\sqrt{5}}{2} + O(\frac{1}{n})$	
Boucle Interne	$\frac{65-29\sqrt{5}}{10}n + \frac{63-28\sqrt{5}}{20} + O(\frac{1}{n})$	$1 + \sqrt{5} + O(\frac{1}{n})$	
MultiBoucle	$\frac{7\sqrt{5}-15}{10}n + \frac{69\sqrt{5}-129}{160} + O(\frac{1}{n})$	$\sqrt{5} + O(\frac{1}{n})$	
Remplissage MultiBoucle	$\frac{5+\sqrt{5}}{2} + O(\frac{1}{n})$	$\frac{\sqrt{5}-1}{2} + O(\frac{1}{n})$	

Fig. 7.7 : Répartition des bases et composition en sous structures d'une structure secondaire pour $e = 1$ et $b = 0$. Le *nombre* des remplissages est le nombre moyen de remplissages par MultiBoucle.

Partie III

GÉNÉRATION ALÉATOIRE DE STRUCTURES SECONDAIRES D'ARN

8. GÉNÉRATION ALÉATOIRE ET BIOINFORMATIQUE

La recherche en génomique a soulevé au cours des deux dernières décennies un nombre important de problèmes algorithmiques et statistiques nouveaux. Dans certains cas, on a pu transposer des outils et théories déjà existants pour modéliser et résoudre ces problèmes mais, dans d'autres cas, il a été nécessaire de créer des concepts nouveaux. Ces concepts ne bénéficient pas toujours de l'expertise qu'avaient acquis des générations de mathématiciens et d'informaticiens dans le domaine, par exemple, de la théorie des langage. On se trouve aujourd'hui dans des situtations où les outils d'analyse manquent, ce qui justifie pleinement l'emploi de séquences aléatoires.

8.1 Analyse des algorithmes heuristiques

De nombreux algorithmes appliqués à la bioinformatiques fonctionnent selon des heuristiques complexes, et leurs performance et correction sont virtuellement impossible à analyser en l'absence d'une chaîne algorithmique complète. Une telle chaîne algorithmique est nécessaire, car les modèles considérés contiennent des paramètres en constante évolution.

Par exemple, l'algorithme de repliement de M.Zucker implémenté dans MFold fait appel à une base de données de valeurs pour l'energie libre des k-boucles. Il est impossible de faire abstraction de ces valeurs pour analyser la complexité de cet algorithme. Or celles ci sont en constantes évolution, *collant* aux energie libres constatées experimentalement. Il est donc aujourd'hui impossible d'analyser la complexité de cet algorithme par des méthodes statiques de type *série génératrices* sans que les résultats obtenus ne soient susceptibles d'être fortement mises en question à la prochaine mise à jour des valeurs thermodynamiques. On retrouve un peu le même type de problème pour les algorithmes de mesure de similarité entre deux séquences utilisés par BLAST ou FASTA, principalement parceque l'algorithme exact est quadratique. Ces outils sont très utilisés par les biologistes du génome, mais on ne sait que peu de choses sur leur sensibilité.

Ces algorithmes prennent une séquence A relativement courte, extraient de A un ensemble de motifs courts qu'ils essaient d'aligner sur une séquence B cible longue. Une fois un *hit* trouvé à un faible nombre de mutations près, on étend l'alignement candidat à partir de l'*ancrage*. La présence de seuils dans ces algorithmes rend critique l'analyse de sensibilité. Des chiffres sont proposés par les concepteurs, mais la méthode d'extension utilisée par ces algorithmes introduirait selon L. Noe[8] un biais, qui amènerait une surestimation de la sensibilité de ces algorithmes. On trouvera en annexe un algorithme de génération aléatoire ayant permis la mise en évidence expérimentale de ce biais.

8.2 Problème de significativité d'un phénomène observé

Ce problème de retard de la recherche théorique sur les modéles utilisés pratiquement est aussi critique dans l'analyse de la significativité d'une donnée observée. Le problème qui se pose est alors le suivant :

> Est ce que la valeur constatée d'un paramètre expérimentalement dans une séquence est supérieure à la valeur moyenne de ce paramètre sur l'ensemble des séquences du modèle ?

Ce problème a été résolu par P. Nicodème, B. Salvy et P. Flajolet dans [18] dans les cas où le modèle est une expression régulière. Ils proposent pour cela une chaîne algorithmique complète implémentée dans *regexp*, un package de Maple. Malheureusement, le formalisme des expressions régulières n'est pas assez expressif pour décrire des modèles plus structurés[1]. Par exemple, le modèle structurel des ARNs n'est pas rationnel, d'après le lemme de l'étoile.

On trouve donc dans la bioinformatique des modèles et des formalismes dont on ne sait pas encore automatiser une étude statistique. On se contente alors d'une analyse expérimentale *in silico* passant par la génération aléatoire de séquences issues du modèle, suivie d'une analyse des séquences obtenues permettant de déduire une approximation de la valeur du paramètre d'intérêt dans le *modèle nul*. On peut alors en déduire la significativité du phénomène observé expérimentalement.

Au contraire, on peut mettre en évidence les manques d'un modèle en observant un écart entre les propriétés constatées expérimentalement des séquences et celles des séquences aléatoires engendrées selon le modèle.

8.3 Inférence de propriétés

La recherche sur le génome confronte ses acteurs à des données de tailles hétérogènes. Par exemple, la taille des promoteurs est de quelques centaines de bases, quand l'ensemble du matériel génétique humain est codé sur 3,5 milliards de bases. On distingue donc les apprentissages pouvant être réalisés par un humain[2] de ceux requérant l'usage d'une machine.

Dans le cas d'une analyse humaine, des séquences aléatoires couplées à un logiciel de visualisation permettent, par exemple, de mettre l'expertise des biologistes du génome au service de la conception de modèles pour les séquences.
On pense en particulier au critère de *crédibilité*, souvent evoqué lorsqu'on montre des représentations de séquences à des acteurs expérimentés de la recherche sur le génome. Ce critère est difficilement formalisable, mais peut servir d'oracle, *stricto sensus*, pour la validation de modèle, ou peut fournir la direction dans laquelle le modèle doit évoluer. On intègre donc ici les séquences aléatoires dans une méthodologie de recherche.

Dans le cas d'une inférence *in silico*, on peut grace à des séquences aléatoires vérifier expérimentalement une propriété de stabilité pour l'algorithme utilisé. On rappelle qu'un algorithme d'apprentissage est un algorithme qui, étant donné des ensembles d'exemples positifs et négatifs, renvoie un modèle contenant toutes les séquences positives et excluant les exemples négatifs. On souhaite en outre qu'un tel algorithme ne soit pas trop spécialisé, c'est à dire qu'il n'accepte pas uniquement les exemples positifs ni ne rejette que les exemples négatifs.
Une propriété qu'on voudrait voir satisfaite pour un tel algorithme est la suivante : Soit M un modèle, si nous engendrons des séquences aléatoirement dans M, et que nous faisons de l'apprentissage positif sur ces séquences, retombe t'on sur le modèle original ou à quelle vitesse s'en éloigne-t-on

[1] Cependant, il y a fort à parier que ce type de démarche d'analyse automatique sera transposé dans un futur proche au cas des langages context-free . . .

[2] D'un point de vue informaticien, un humain est une machine implémentant un algorithme d'inférence incroyablement puissant, mais doté d'une mémoire ridicule

? Dans le domaine des séquences, on peut considérer qu'un modèle est entre $\emptyset$ et V^*, un *excellent* moteur d'inférence de modèle serait alors un moteur qui, pour un nombre borné de séquence aléatoires, ne convergerait expérimentalement ni vers $\emptyset$, ni vers V^*. Plus pragmatiquement, on peut utiliser des séquences aléatoire pour calibrer les paramètres d'un moteur d'inférence, en vérifiant pour quelles valeurs des paramètres les itérations successives du moteur sur un modèle s'éloignent pas trop vite du modèle initial …

9. TECHNIQUES DE GÉNÉRATION ALÉATOIRE

On va présenter ici diverses techniques génériques de génération aléatoires, classées par complexité (et par chronologie) décroissante. On remarque que la génération aléatoire sur une famille d'objets combinatoires peut toujours se ramener à une génération de mots d'un langage[1]. On pourra, par exemple, s'intéresser au codage machine des mots à engendrer dans l'implémentation d'un algorithme.

Dans un premier temps, on s'intéresse à une génération assez simple, la génération aléatoire itérative, qu'on applique naturellement à la génération aléatoire de chemins. On l'applique en annexe à la génération aléatoire de chemins culminants, des chemins de langages associés *associés* non algébriques.

Nous présenterons ensuite les principes de l'approche récursive avec dénombrement préliminaire. Nous prouverons sa correction, à partir du moment où une étape de dénombrement préliminaire a eu lieu.

Nous finirons sur la génération de Boltzmann due à Duchon, Flajolet, Louchard et Schaeffer, et qui permet une génération aléatoire en temps linéaire de séquences pour une large catégorie de langages non contextuels, y compris celui des structures secondaires d'ARN. Le prix à payer pour la mise en oeuvre d'une telle technique est un léger relâchement sur la taille des objets générés.

9.1 Approche itérative

Dans [26], H. Wilf propose une formalisation du concept d'objet combinatoire. Il propose une modélisation basée sur un graphe orienté acyclique doté un état initial et d'un état final. Il numérote les arêtes sortant d'un sommet. Il code alors la structure engendrée par un cheminement entre l'état initial et l'état final par une séquence de nombres correspondant aux numéros des arêtes empruntées pour sortir des états. Il propose enfin un algorithme postulant une possibilité de dénombrement des objets issus d'un état pour pratiquer une génération aléatoire uniforme.

Nous transposons ici ce concept aux langages, qui sont rationnels quand on considère leurs restriction aux mots d'une taille n.

Théorème 20 (Génération aléatoire itérative) :
Soit $\mathcal{L}$ un langage sur l'alphabet $S = \{s_1, s_2, \ldots, s_m\}$. Soit $\mathcal{L}_n$ la restriction de cet ensemble aux mots de taille n.
Soit ω_p le préfixe d'un mot de $\mathcal{L}_n$, on appelle $p_s(\omega_p)$ la probabilité que w_p soit suivi par $s \in S$ dans un mot de $\mathcal{L}_n$.

$$p_s(\omega_p) = \frac{|\{\omega' | \omega_p.s.\omega' \in \mathcal{L}_n\}|}{|\{\omega' | \omega_p.\omega' \in \mathcal{L}_n\}|}$$

Alors l'algorithme suivant engendre bien aléatoirement un mot de taille n pris uniformément sur $\mathcal{L}_n$:

[1] Pas nécessairement non contextuel…

- $w = \varepsilon$
- *Tant que* $|w| < n$*, prolonger* w *par une de* X *avec probabilités* $p_{x_1}(w), \ldots, p_{x_m}(w)$

Preuve : *En effet, partant du préfixe* ϵ*, on le prolonge par des lettres* c_1*,* c_2*,* $c_3, \ldots, c_n$ *choisies selon les probabilités* $p_{c_1}(\epsilon)$*,* $p_{c_2}(c_1)$*,* $p_{c_3}(c_1.c_2), \ldots, p_{c_n}(c_1.c_2.\cdots.c_{n-1})$*. La probabilité d'émission d'un mot* $\omega = c_1.c_2.c_3.\cdots.c_n$ *est donc égale à :*

$$\begin{aligned} p(c_1.\ldots.c_n) &= p_{c_1}(\epsilon)p_{c_2}(c_1)p_{c_3}(c_1.c_2)\cdots p_{c_n}(c_1.c_2.\cdots.c_{n-1}) \\ &= \frac{|\{\omega'|c_1.\omega' \in \mathcal{L}_n\}|}{|\mathcal{L}_n|} \frac{|\{\omega'|c_1.c_2.\omega' \in \mathcal{L}_n\}|}{|\{w'|c_1.w' \in \mathcal{L}_n\}|} \cdots \frac{|\{\omega'|c_1,\ldots,c_n.\omega' \in \mathcal{L}_n\}|}{|\{\omega'|c_1,\cdots,c_{n-1}.\omega' \in \mathcal{L}_n\}|} \\ &= \frac{|\{\omega'|c_1.\omega' \in \mathcal{L}_n\}|}{|\{\omega'|c_1.\omega' \in \mathcal{L}_n\}|} \frac{|\{\omega'|c_1.c_2.\omega' \in \mathcal{L}_n\}|}{|\{w'|c_1.c_2.w' \in \mathcal{L}_n\}|} \cdots \frac{|\{\omega'|c_1,\ldots,c_n.\omega' \in \mathcal{L}_n\}|}{|\mathcal{L}_n|} \\ &= \frac{1}{|\mathcal{L}_n|} \text{ car } \{\omega'|c_1, \ldots, c_n.\omega' \in \mathcal{L}_n\} = \epsilon \end{aligned}$$

Tous les mots de L_n *étant choisi par ce processus avec une probabilité* $\frac{1}{|L_n|}$*, ce procédé de génération aléatoire est donc uniforme.*

Touver un algorithme de génération aléatoire uniforme de mots de ce langage revient donc à compter les suffixes des mots du langage étudié. Bien sûr, il n'est pas question de calculer les probabilités pour tous les couples préfixe/symbole, car le nombre de préfixes d'un langage peut être exponentiel. Cependant, on peut dans de nombreux cas regrouper les préfixes en classes.

On illustre ce phénomène de regroupement en classes par une application aux chemins de Dyck bilatères.
On rappelle que les chemins de Dyck bilatères $\mathcal{D}$ sont les chemins du demi plan d'abscisses positives qui partent de $(0, 0)$ et arrivent en $(2n, 0)$ avec des pas $(+1, +1)$ et $(+1, -1)$. Ils sont en bijection avec les mots ω sur $\{a, b\}^*$ tels que $|\omega|_a = |\omega|_b$ par un codage trivial illustré par la Figure

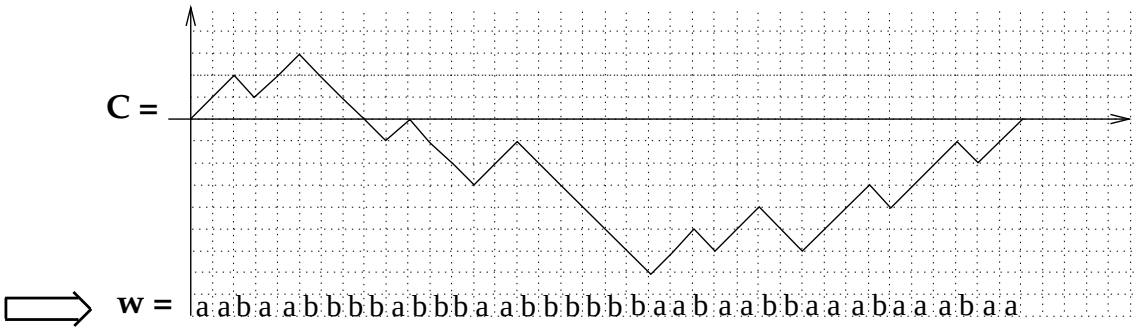


Fig. 9.1 : Exemple de chemin de Dyck bilatère et codage par un mot de $\{a, b\}^*$

9.1.
Soit ω_p un préfixe du mot associé à un chemin de $\mathcal{D}_{2n}$, et soit $\psi(\omega_p) = |\omega_p|_a - |\omega_p|_b$ l'altitude finale de ce préfixe. Alors $|\{\omega'|\omega_p.s.\omega' \in \mathcal{D}_{2n}\}|$ est uniquement déterminé par $\psi(\omega_p)$. En effet, les suffixes dans $\mathcal{D}_{2n}$ des chemins finissant en $\psi(\omega_p)$ sont exactement les chemins allant de $\psi(\omega_p)$ à 0 en $|2n - w_p|$ pas. Cet ensemble est indépendant du chemin parcouru par w_p.
De plus, on remarque que l'ensemble des chemins allant de $\psi(\omega_p)$ à 0 en $\kappa := 2n - |w_p|$ pas peut être vu comme le choix de $\zeta := \frac{\kappa - \psi(\omega_p)}{2}$ pas montants parmis κ pas. On obtient donc naturellement une expression binômiale du nombre $C_{j\ k}$ de chemins de j à 0 en k pas :

$$C_{j\ k} = \binom{k}{\frac{j+k}{2}}$$

$$\Rightarrow |\{\omega' | \omega_p.a.\omega' \in \mathcal{D}_{2n}\}| = C_{\psi(\omega_p)+1,\ 2n-|\omega_p|-1} = \binom{2n-|\omega_p|-1}{\frac{\psi(\omega_p)+2n-|\omega_p|}{2}}$$

$$\Rightarrow p_a(\omega_p) = \frac{C_{\psi(\omega_p)+1,\ 2n-|\omega_p|-1}}{C_{\psi(\omega_p),\ 2n-|\omega_p|}} = \frac{\binom{2n-|\omega_p|-1}{\frac{\psi(\omega_p)+2n-|\omega_p|}{2}}}{\binom{2n-|\omega_p|}{\frac{\psi(\omega_p)+2n-|\omega_p|}{2}}} = \frac{2n-|w_p|-\psi(\omega_p)}{4n-2|w_p|}$$

On peut donc, en un nombre constant d'opération arithmétiques, déterminer à chaque itération par quel type de pas prolonger un chemin de Dyck bilatère construit itérativement. Donc on dispose d'un algorithme de complexité linéaire pour la génération de mots de Dyck bilatères.

Cependant, il est rare qu'on puisse trouver une factorisation aussi indépendante du contexte, et dont on peut établir une expression du cardinal. Une étape de précalcul est donc souvent nécessaire pour évaluer ces cardinaux à partir de récurrences.

9.2 Le cas des structures décomposables

Dans [10], Flajolet, Zimmermann et Van Cutsem proposent un algorithme de génération uniforme pour une classe infinie de modèles de structures décomposables. L'algorithme prend en entrée une spécification de structure combinatoire qui est une décomposition hiérarchique des objets d'intérêt. Il transpose alors cette spécification dans une forme normale, dans laquelle des relations de récurrences apparaissent simplement entre les sous structures de la décomposition récursive. On peut alors dénombrer les objets pour différentes tailles, ce qui permet par un procédé analogue à celui de Wilf de générer uniformément un objet répondant à cette spécification.

On va illustrer ces principes complexes, mais *complètement automatisés* et implémentés dans le package Maple *combstruct*, ou bien dans GenRGenS, en s'intéressant à un sous ensemble des structures décomposables : les langages engendrés par des grammaires non contextuelles.

9.3 Génération à partir d'une grammaire non contextuelle

9.3.1 Définitions préliminaires

Définition 23 (Grammaire non contextuelle) :
Une grammaire non contextuelle est caractérisée par la donnée d'un quintuplet $G = (I, T, NT, \sigma)$ où :

- *$I \in NT$ est le symbole terminal initial.*
- *T est un ensemble des symboles terminaux.*
- *NT est un ensemble de symboles non terminaux.*
- *σ est un ensemble de règle R de réécriture de la forme :*

$$R \ \rightarrow \ S_1 \ S_2 \ \ldots \ S_k$$

On peut formaliser la notion de règle en $R \in NT \times (NT \cup T)^*$.[2]

[2] On pourra s'amuser du fait que l'ensemble des grammaires non contextuelles sur des vocabulaire finis est décrit par une expression rationnelle.

Le rapport entre grammaire et langage est lié à la notion de réécriture. On associe à une grammaire le langage des mots sur le vocabulaire terminal obtenus par réécritures successives des non terminaux à partir du non terminal initial.
On introduit ensuit une forme qui permet une manipulation algorithmique efficace des grammaires : la forme normale de Chomsky.

Théorème 21 (Forme Normale de Chomsky) :
Toute grammaire non contextuelle $G = (I, T, NT, \sigma)$ admet une grammaire engendrant le même langage dont les règles sont de la forme :

$$R \to \varepsilon$$
$$R \to c$$
$$R \to S.T$$

avec $c \in T$, $(R, S, T) \in NT$ et $I \in NT$ est le symbole non terminal initial de la grammaire.

Preuve : *Très brièvement, on casse les règles du type $R \to S_1.S_2.\ldots.S_k$ en introduisant des non terminaux $N_1, N_2, \ldots, N_{k-2}$ tels que :*

$$R \to S_1.S_2 \ldots S_k \Rightarrow \left\{ \begin{array}{l} R \to \phi(S_1).N_1 \\ N_1 \to \phi(S_2).N_2 \\ \vdots \\ N_{k-2} \to \phi(S_{k-1}).\phi(S_k) \end{array} \right\}$$

Où $\phi(S) = S$ si $S \in NT$ et $\phi(S) = X_S$ si $S \in T$. Et, pour chaque terminal c, on ajoute une règle $X_c \to c$.

Remarques : Il existe au plus une règle dans la grammaire de la forme $R \to \varepsilon$. De plus, on peut, en rendant la grammaire *propre* avant de la mettre en forme normale, faire en sorte que R ne soit jamais utilisé dans la définition d'un autre non terminal que I initial.

9.3.2 Phase de dénombrement

A partir d'une grammaire non ambiguë en forme normale de Chomsky , on peut sans difficulté compter les objets d'une taille n engendrables par des réécritures d'un non terminal. pour cela on adopte une stratégie récursive.
Soit r une règle et, pour S un symbole non terminal, $s[n]$ est le nombres de mots de taille n issus de S :

$$r = A \to B.C \quad \Rightarrow \quad a[n] := \sum_{i=1}^{n-1} b[i]c[n-i] \tag{9.1}$$

$$r = A \to B_1 + B_2 + \ldots + B_2 \quad \Rightarrow \quad a[n] := \sum_{i=1}^{k} b_i[k] \tag{9.2}$$

$$r = A \to c \in T \quad \Rightarrow \quad a[n] := \left\{ \begin{array}{ll} 1 & \text{si } n = 1 \\ 0 & \text{sinon} \end{array} \right. \tag{9.3}$$

$$r = A \to \varepsilon \quad \Rightarrow \quad a[n] := \left\{ \begin{array}{ll} 1 & \text{si } n = 0 \\ 0 & \text{sinon} \end{array} \right. \tag{9.4}$$

Remarques : En réalité, le dénombrement est calculé itérativement pour i allant de 0 à n. Les résultats intermédiaire sont stockés dans une table.
D'autre part, on évite les affectations récursives dans le produit ($a_n := a[n]b[0]$) en remarquant qu'il existe un seul non terminal susceptible de dériver un mot de taille nul et qu'il n'est pas utilisé dans la définition d'un produit (voir remarque précédente), on a donc toujours $b[0] = 0$.

9.3.3 Génération

Une fois la phase de dénombrement préliminaire achevée, on applique l'algorithme 9.3.3.

Algorithme 1 Génère aléatoirement uniformément un mot d'une grammaire non contextuelle en forme normale de Chomsky

1: $\omega \leftarrow (I, n)$
2: **Tant Que** $\omega \notin (T \times \{1\})^*$ **Faire**
3: $\omega = \omega_1.(A, m).\omega_2$ tel que $A \in NT$
4: **Si** $A \rightarrow B_1 + B_2 + \ldots + B_k$ **Alors**
5: Choisir j, $1 \leq j \leq k$ au hasard tel que $p(j, m) = \frac{b_j[m]}{a[m]}$
6: $\omega \leftarrow \omega_1.(B_j, m).\omega_2$
7: **Sinon Si** $A \rightarrow B.C$ **Alors**
8: Choisir i, $0 < i < m$ tel que $p(i, m) = \frac{b[i].c[m-i]}{a[m]}$
9: $\omega \leftarrow \omega_1.(B, i).(C, m - i).\omega_2$
10: **Sinon Si** $A \rightarrow c \in T$ **Alors**
11: $\omega \leftarrow \omega_1.(c, 1).\omega_2$
12: **Fin Si**
13: **Fin Tant Que**

Preuve : *On va prouver qu'un tel algorithme est bien un générateur uniforme du langage par récurrence sur le nombre maximum de réécritures nécessaires pour réécrire le mot à partir d'un non terminal A:*

- **Pour** $n = 1$ **:**
  Les mots nécessitant une réécriture sont bien engendrés uniformément. En effet, la seule règle applicable étant $A \rightarrow c$, le mot engendré est le seul mot du langage, or il est engendré avec probabilité 1.

- **Pour** $n < k$ **:**
  On suppose que l'algorithme engendre uniformément les différents mots nécessitant moins de k réécritures.

- **Pour** $n = k$ **:**
  On sépare l'étude des différentes règles :

  - $A \rightarrow B_1 + B_2 + \ldots + B_k$ *:*
    L'algorithme, appliqué à $B_1, B_2, \ldots$ vérifie l'hypothèse de récurrence, car le nombre de réécritures nécessaires pour engendrer les mots issus de ces non terminaux est inférieur à k. Donc les mots issus de B_j, sont tirés avec probabilités $\frac{1}{b_j[k]}$ à partir de B_i. D'où, soit $p(\omega)$ la probabilité d'un mot de B_j soit issu de A :

$$p(\omega) = \frac{b_j[k]}{a[k]} \frac{1}{b_j[k]} = \frac{1}{a[k]}$$

  - $A \rightarrow B.C$ *:*
    Idem pour B et C, qui vérifient les conditions de l'hypothèse de récurrence. La probabilité qu'un mot de taille i(resp. $k - i$) soit engendré par une série de réécritures de B (resp. C) est $\frac{1}{b[i]}$ (resp. $\frac{1}{c[k-i]}$). La probabilité qu'un mot de taille k soit obtenu par concaténation d'un mot de

taille i issu de B et d'un mot de taille $k-i$ issu de C est égale à $\frac{1}{b[i]c[k-i]}$. Donc la probabilité d'émission d'un mot ω issu de A par une répartion $(i, k-i)$ des tailles entre B et C est :

$$p(\omega) = \frac{1}{b[i]c[k-i]} \frac{b[i].c[k-i]}{a[k]} = \frac{1}{a[k]}$$

Donc, si la grammaire est non ambiguë [3], et que $a[k]$ contient bien le nombre d'objet issus de A de taille k, alors la génération est uniforme.

On dispose donc d'un algorithme de génération aléatoire uniforme. De plus, sa complexité pour des séquences de tailles n est en $O(n^2)$ pour le précalcul, en $O(nlog(n))$ pour la génération. Cependant, ce type de génération est insuffisant dans notre optique de validation de modèle. En effet, les modèles biologiques ne sont que rarement uniquement structurels, ils présentent aussi des critères statistiques qui brisent l'uniformité pour coller à une réalité. L'algorithme qui suit prend en compte de la distribution des symboles terminaux.

9.4 Génération non uniforme pondérée de mots d'un langage non contextuel

9.4.1 Apparition des poids

Dans [4], A. Denise, O. Roques et M. Termier présentent une génération aléatoire non uniforme contraignant l'espérance du nombre d'occurences des symboles terminaux. Pour cela, ils proposent de pondérer les symboles terminaux, ce qui impose une modification des règles de dénombrement récursif.

En effet, dans la génération uniforme on peut considérer qu'il est affecté à chaque mot du langage un poids égal à 1. Or, la contribution d'un mot aux termes $a[n]$ de la phase de dénombrement est égale à 1, mais aussi au produit des contribution des symboles terminaux. On observe cette propriété en développant tous les non terminaux utilisés dans la définition du non terminal jusqu'à n'obtenir que des mots de T^*. Soit φ_n la fonction qui compte les mots de taille n issus de $NT \cup T$:

$$R \to S + T \to \ldots \to \omega_1 + \omega_2 + \omega_3 + \ldots + \omega_{r[n]}$$

$$\varphi_n(R) = \varphi_n(S) + \varphi_n(T) = \ldots = \varphi_n(\omega_1) + \varphi_n(\omega_2) + \varphi_n(\omega_3) + \ldots + \varphi_n(\omega_{r[n]})$$

En appliquant alors la relation (9.1), étendue naturellement au produit de plus de deux opérandes, on obtient, en se retenant d'appliquer (9.3), une expression du poids d'un mot. De plus, $\omega \in \{s_1, s_2, \ldots, s_k\}$, donc soit $\omega[m]$ la m-ième lettre d'un mot :

$$\varphi_n(R) = \sum_{i=1}^{r[n]} \prod_{j=1}^{n} \varphi_1(\omega_i[j]) = \sum_{i=1}^{r[n]} \varphi_1(s_1)^{|\omega_i|_{s_1}} \ldots \varphi_1(s_k)^{|\omega_i|_{s_k}}$$

$$\varphi_n(R) = \sum_{i=1}^{r[n]} \prod_{j=1}^{k} \varphi_1(s_j)^{|\omega_i|_{s_j}}$$

On voit donc très clairement apparaître une fonction susceptible de pondérer les mots de la génération. On visualise aussi facilement que les probabilités d'émission des mots seront directement affecté par une modification de cette fonction.

[3] Une grammaire peut être ambiguë intrinsèquement ou maladroitement

9.4.2 Adaptation de l'algorithme

On décide donc de modifier la règle de dénombrement (9.3) en :

$$r = A \to c \in T \quad \Rightarrow \quad a[n] := \begin{cases} \pi(c) & \text{si } n = 1 \\ 0 & \text{sinon} \end{cases}$$

Où π est un fonction de T dans $\mathbb{R}$, qu'on étend à un mot $\omega \in \{s_1, s_2, \ldots, s_k\}^*$ puis à un langage $\mathcal{L}$:

$$\pi(\omega) = \prod_{s \in \{s_1,\ldots,s_k\}} \pi(s)^{|\omega|_s}$$

$$\pi(\mathcal{L}) = \sum_{\omega \in \mathcal{L}} \pi(\omega)$$

Alors l'algorithme modifié génère un mot ω de $\mathcal{L}$ avec une probabilité $p(\omega)$ telle que :

$$p(\omega) = \frac{\pi(\omega)}{\pi(L)}$$

Soit la série génératrice suivante :

$$L_\pi(z, x_1, \ldots, x_k) = \sum_{\omega \in (L)} \pi(\omega) z^{|\omega|} x_1^{|\omega|_{s_1}} \ldots x_k^{|\omega|_{s_k}}$$

Alors l'espérance $\mathbb{E}(N_s)$ du nombre de symboles s dans un mot engendré de taille n est telle que :

$$\mathbb{E}(N_s) = \frac{\sum_{\omega \in \mathcal{L}_n} \pi(\omega)|\omega|_s}{\pi(\mathcal{L}_n)} = \frac{[z^n]\frac{\partial L_\pi(z,x_1,\ldots,x_k)}{\partial s}(z, 1, \ldots, 1)}{[z^n]L_\pi(z, 1, \ldots, 1)}$$

Les auteurs montrent ensuite qu'il existe deux classes pour lesquelles on peut *à priori* fixer une distribution des symboles terminaux, puis calculer des poids garantissant cette distribution à l'asymptotique : Les langages rationnels et certains langages non contextuelles.

9.4.3 Relation pondération/distribution

Pour les langages rationnels, il est proposé une démarche qui tient compte de la stabilité de la distribution quand on multiplie toutes les pondérations par une constante. On peut alors imposer un pôle de plus petit module égal à 1, ce qui simplifie l'étude asymptotique, et donc permet de construire un système d'équations déterminant les pondérations.

Le cas qui nous intéresse tout particulièrement est celui des grammaires non contextuelles. En effet, nous les avons utilisées pour modéliser les structures secondaires d'ARNs, et nous avons constaté, comme l'analyse des comportements asymptotique le laisse supposer, des écarts très importants entre les séquences engendrées uniformément et celles analysées dans une base de donnée. En particulier, les sous structures sont beaucoup trop nombreuses, et beaucoup trop courtes. Il est donc important, afin de générer des séquences *crédibles*, de pouvoir contraindre les nombres d'occurences des symboles terminaux. Pour cela A. Denise, O. Roques et M. Termier adaptent un théorème de Drmota, issu de [5] :

Théorème 22 (Asymptotique des proportions de terminaux) :
Soient $f_1, f_2, \ldots, f_N$ les séries génératrices associés aux N symboles non terminaux d'une grammaire.
Soient $(f_i = F_i(z, x_1, \ldots, x_k, f_1, \ldots, f_N))_{1 \le i \le N}$ le système d'équation associé aux règles d'une grammaire non contextuelle, où les F_i sont des polynômes en leurs différentes variables.
Soit $\mathbf{F}(z, x_1, \ldots, x_k, y_1, \ldots, y_N) = (\frac{\partial F_i}{\partial y_j}(z, x_1, \ldots, x_k, y_1, \ldots, y_N))_{1 \le i,j \le N}$.
Si les conditions suivantes sont remplies :

1. $(F_1(0, x_1, x_k, f_1, f_N), \ldots, F_N(0, x_1, x_k, f_1, f_N)) = (0, 0, \ldots, 0)$
2. $(F_1(z, x_1, x_k, 0, \ldots, 0), \ldots, F_N(z, x_1, x_k, 0, \ldots, 0)) \neq (0, 0, \ldots, 0)$
3. $\exists (i, j, k) \in [1, N]^3, \quad \dfrac{\partial F_i}{\partial f_j \partial f_k} \neq 0$
4. *Il existe un ensemble de N cônes $(k+1)$-dimensionnels $C_i \subseteq \mathbb{R}^{k+1}$ tel que, pour tout $(n, m_1, \ldots, m_k) \in C_i$ suffisamment lointain, le coefficient de terme $z^n x_1^{m_1} \ldots x_2^{m_2}$ de la série génératrice du langage soit non nul.*
5. *Le graphe de dépendance de la grammaire ayant pour sommets les non terminaux est fortement connexe.*
6. *Il existe une solution positive $(y'_1, y'_2, \ldots, y'_N, z')$ au système d'équations :*

$$\begin{cases} y'_1(z', 1, \ldots, 1) & = & F_1(z', 1, \ldots, 1, y_1, \ldots, y_N) \\ \vdots & \vdots & \vdots \\ y'_N(z', 1, \ldots, 1) & = & F_N(z', 1, \ldots, 1, y_1, \ldots, y_N) \\ 0 & = & \det(I - \mathbf{F}(z, 1, \ldots, 1, y_1, \ldots, y_N)) \end{cases}$$

Soit $(y_1, y_2, \ldots, y_N, z(x_1, \ldots, x_k))$ une solution de :

$$\begin{cases} y_1(z, x_1, \ldots, x_k) & = & F_1(z, x_1, \ldots, x_k, y_1, \ldots, y_N) \\ \vdots & \vdots & \vdots \\ y_N(z, x_1, \ldots, x_k) & = & F_N(z, x_1, \ldots, x_k, y_1, \ldots, y_N) \\ 0 & = & \det(I - \mathbf{F}(z, x_1, \ldots, x_k, y_1, \ldots, y_N)) \end{cases}$$

telle que $(y_i(z, 1, \ldots, 1) = y'_i)_{1 \leq i \leq N}$, alors la proportion asymptotique μ_{s_i} de symboles terminaux s_i dans un mot de la grammaire est donnée par :

$$\mu_{s_i} = -\frac{1}{z(1, \ldots, 1)} \frac{\partial z}{\partial x_i}(1, \ldots, 1)$$

L'espérance $\mathbb{E}(N_i)$ du nombre de symboles s_i est donc telle que $\mathbb{E}(N_i) = n\mu_{s_i} = -\frac{n}{z(1,\ldots,1)} \frac{\partial z}{\partial x_i}(1, \ldots, 1)$
.

Les conditions d'application de ce théorème sont un peu abruptes, on essaye de les traduire dans le langage *courant* :

1. $\Rightarrow$ Aucun non terminal ne génère le mot vide.
2. $\Rightarrow$ Il existe des termes constants dans la grammaire. Condition nécessaire pour que la grammaire puisse dériver un mot du vocabulaire terminal.
3. $\Rightarrow$ Ce système n'est pas linéaire sur les séries génératrices de non terminaux. S'il l'était, le langage engendré serait rationnel.
4. $\Rightarrow$ Condition d'apériodicité de la grammaire. Condition imposée pour éviter un comportement oscillant de l'asymptotique, voir carrément pas de convergence.
5. $\Rightarrow$ Selon Drmota [5], si cette condition est violée, alors on peut voir apparaître différents types de distribution, ce qui fausse l'assertion sur la distribution.

6. $\Rightarrow$ Sert principalement d'étalon pour déterminer la bonne solution générale, car l'élimination est toujours possible dans ce type de système.

On évoquera une utilisation de ce théorème dans la section suivante, et en annexe sous la forme d'un fichier Maple.

On dispose donc, une fois dépassées quelques lourdeurs administratives, d'un outil permettant la génération aléatoire en fréquence contrainte. Cependant, son implémentation au sein de GenRGenS a prouvé que sa complexité quadratique temps/mémoire est atteinte en pratique. On souhaite donc faire appel à d'autres algorithmes de complexité moyenne si possible linéaire.

9.5 Génération de Boltzmann : Principe

Le principe *philosophique* sous jacent à la génération de Boltzmann est : Si, au lieu d'imposer une taille fixe n aux objets engendrés, on s'autorise une variation relative d'un ε autours de n, alors on sait alors générer des mots en temps linéaire pour une classe très importante d'objets combinatoire, qui semble contenir les langages algébriques.
On va très brièvement présenter la méthode de Boltzmann. Dans [6], les auteurs donne des méthodes pour construire des générateurs de Boltzmann sur des structures définies par des opérateurs d'union et de produit dans l'univers non étiqueté et d'union, de produit, de cycle, d'ensemble et de séquences dans l'univers étiqueté.
On ne décrira ici que le traitement des opérateurs d'union et de produit non étiqueté, qui suffisent pour définir les grammaires non contextuelles. On définit un générateur de Bolzmann de paramètre x pour $\mathcal{C}$ comme le montre l'algorithme 2.

Algorithme 2 Boltzmann($\mathcal{C}$,x)

Si $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ **Alors** {**Produit Carthésien**}
  Renvoyer Boltzmann($\mathcal{A}$,x).Boltzmann($\mathcal{B}$,x)
Sinon Si $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ **Alors {Union disjointe}**
  $p_a \leftarrow \frac{A(x)}{C(x)}$
  Tirer $0 \leq \rho < 1$ **aléatoirement**
  Si $\rho < p_a$ **Alors**
    Renvoyer Boltzmann($\mathcal{A}$,x)
  Sinon
    Renvoyer Boltzmann($\mathcal{B}$,x)
  Fin Si
Sinon {Atome : $\mathcal{C} = c$**}**
  Renvoyer c
Fin Si

Théorème 23 (Probabilité d'une structure sous Boltzmann) :
Soit $C(z)$ la série génératrice d'un ensemble $\mathcal{C}$ de structures combinatoires spécifié en utilisant uniquement des sommes et des unions.
Alors la probabilité p_ω d'émission d'un mot $\omega \in \mathcal{C}$ de taille $|\omega| = n$ par l'algorithme ci dessus pour $\mathcal{C}$ de paramètre x est telle que :

$$p_\omega = \frac{x^n}{C(x)}$$

Preuve : *Par récurrence sur le nombre k de récursion de Boltzmann($\mathcal{C}$,x) nécéssaires pour engendrer ω :*

- $k = 0$ **:**
  $\omega = c \Rightarrow p_\omega = 1$ *conformément au modèle proposé.*

- **Hypothèse :**
  Pour $k \leq m$, un mot $\omega' \in \mathcal{A}$ engendré en moins de k itération l'a été avec probabilité $p_{\omega'} = \frac{x^{|\omega'|}}{A(x)}$.

- **Pour $\mathbf{k = m + 1}$:**
  $\mathcal{C} = \mathcal{A} \times \mathcal{B} \Rightarrow \omega = \omega_1.\omega_2$. *De plus, le nombre de récursions nécessaires à l'algorithme pour engendrer ω_1, ω_2 est $\leq m$:*

$$\Rightarrow p_{\omega_1} = \frac{x^{|\omega_1|}}{A(x)} \; et \; p_{\omega_2} = \frac{x^{|\omega_2|}}{B(x)}$$

$$\Rightarrow p_\omega = p_{\omega_1} p_{\omega_2} = \frac{x^{|\omega_1|} x^{|\omega_2|}}{A(x)B(x)} = \frac{x^{|\omega|}}{C(x)}$$

Car $C(z) = A(z)B(z)$. $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$, union supposée disjointe :

$$\omega = \omega_1 \in \mathcal{A} \Rightarrow p_\omega = \frac{A(x)}{C(x)} p(\omega_1) = \frac{A(x)}{C(x)} \frac{x^{\omega_1}}{A(x)} = \frac{x^\omega}{C(x)}$$

$$\omega = \omega_2 \in \mathcal{B} \Rightarrow p_\omega = (1 - \frac{A(x)}{C(x)}) p(\omega_2) = \frac{B(x)}{C(x)} \frac{x^{\omega_1}}{B(x)} = \frac{x^\omega}{C(x)}$$

Car $C(z) = A(z) + B(z) \Rightarrow 1 - \frac{A(x)}{C(x)} = \frac{C(x) - A(x)}{C(x)} = \frac{B(x)}{C(x)}$.

On remarque que si les unions sont effectivement disjointes, alors le nombre de dérivations nécessaire pour engendrer ω est unique. Donc les objets engendrés le sont suivant la loi énoncée ci dessus.

En particulier, la taille des objets engendrés a des propriétés directement déductibles de la série génératrices de la classe considérée.

Théorème 24 (Espérance et variance de la longueur des objets engendrés) :
Soit $\mathcal{C}$ une classe d'objets, de série génératrice $C(z)$.
Soit N_x la variable aléatoire associée à la longueur d'un objet engendré par une génération de Boltzmann de paramètre x.

$$\mathbb{E}(N_x) = x \frac{\frac{\partial C(z)}{\partial z}(x)}{C(x)}$$

$$\mathbb{E}({N_x}^2) = \frac{x^2 \frac{\partial^2 C(z)}{\partial^2 z}(x) + x \frac{\partial C(z)}{\partial z}}{C(x)}$$

On appelle génération par rejet anticipé appliqué au générateur de Bolzmann l'adjonction d'une variable globale qui contient à chaque instant la somme des tailles des sous structures engendrées, couplée à un mécanisme d'interruption si cette variable dépasse une borne.

Théorème 25 (Linéarité de la génération en taille approximative) :
Soit $\mathcal{C}$ une classe d'objets combinatoires, de série génératrice $C(z)$.
Soit ρ la singularité dominante de $C(z)$.
Soit $0 < \varepsilon < 1$ une tolérance relative sur la taille de l'objet généré.
Si $C(z)$ remplit la condition de moyenne :

$$\lim_{x \to \rho^-} \mathbb{E}(N_x) = +\infty$$

et de variance :

$$\lim_{x \to \rho^-} \frac{\sqrt{\mathbb{E}({N_x}^2) - \mathbb{E}(N_x)^2}}{\mathbb{E}(N_x)} = 0$$

Alors le nombre de tirages nécessaires pour obtenir un objet de taille comprise dans $[(1-\varepsilon)n, (1+\varepsilon)n]$ *tend vers* 1 *quand* $n \to \infty$

10. APPLICATION DES DIFFÉRENTES GÉNÉRATIONS ALÉATOIRES

Dans un premier temps, nous allons utiliser la méthode récursive pour engendrer des séquences aléatoires uniformément. Pour cela, nous allons utiliser deux approches : l'approche récursive et la génération de Boltzmann.

Nous discuterons en suite de la disparité des compositions en sous structures entre les séqences d'ARN ribosomaux et séquences générées aléatoirement uniformément. Nous verrons alors apparaître, autant théoriquement qu'*expérimentalement*, la nécessité d'une génération aléatoire non uniforme.

Nous appliquerons alors alors le théorème de Drmota présenté dans le chapitre précedent à un cas simple, puis nous discuterons des difficultés de l'application du théorème de Drmota au cas la grammaire proposée précédemment.

10.1 Génération uniforme

10.1.1 Approche récursive

Le principe de l'approche récursive et sa justification théorique ayant été présentés au chapitre précédent, on ne reviendra pas dessus. Il reste cependant à concevoir une grammaire non contextuelle adaptée à une génération aléatoire selon ces principes.

Grammaire simple des struc. sec. d'ARN

On propose la grammaire de la figure 10.3.

```
ARN  ->  S | ε
  S  ->  a S b S | a S b | c S | c
```

Fig. 10.1 : Grammaire non contextuelle simple adaptée à la génération uniforme.

On obtient cette grammaire directement à partir de la grammaire *non contextuelle* de Vauchaussade de Chaumont et Viennot[2] mise en forme. En effet, soit G une grammaire non contextuelle.Si, dans la partie droite d'une règle r de G, on substitue à un non terminal N toutes les parties droites des règles dont N est partie gauche, créant ainsi de nouvelles disjonctions pour r, alors on obtient une grammaire G' d'expressivité équivalente. On applique cette propriété pour faire disparaître le `(ARN-ε)` de la grammaire[2] :

```
ARN -> a (ARN-ε) b ARN | c ARN | ε
```

$$\Leftrightarrow \left\{ \begin{array}{l} \texttt{ARN -> S | ε} \\ \texttt{S -> a ARN b ARN | c ARN} \end{array} \right\}$$

$$\Leftrightarrow \left\{ \begin{array}{rcl} \texttt{ARN} & \texttt{->} & \texttt{S} \mid \varepsilon \\ \texttt{S} & \texttt{->} & \texttt{a S b S} \mid \texttt{a S b} \\ & \mid & \texttt{c S} \mid \texttt{c} \end{array} \right\}$$

On a choisi la grammaire la plus simple possible, car, soit k la taille de la grammaire, la génération par approche récursive de mots de taille n est de complexité $O(k'n^2)$ en espace et en temps, avec $k' > k$ la taille de la grammaire mise en forme normale de Chomsky.

Tailles minimales des sous structures

On peut aussi rapidement obtenir une grammaire simples qui impose un nombre minimal de bases dans des boucles terminales ou des empilements. Pour cela, on duplique le symbole S en un symbole T doté des mêmes règles, et donc du même langage engendré. On peut donc substituer T à S dans n'importe quelles partie droite de règle. On fait alors en sorte que les mots issus de T soient exactement ceux encadrés par un couple `a b` :

$$\begin{array}{rcl} \texttt{ARN} & \texttt{->} & \texttt{S} \mid \varepsilon \\ \texttt{S} & \texttt{->} & \texttt{a T b S} \mid \texttt{a T b} \mid \texttt{c S} \mid \texttt{c} \\ \texttt{T} & \texttt{->} & \texttt{a T b S} \mid \texttt{a T b} \mid \texttt{c S} \mid \texttt{c} \end{array}$$

Donc une réécriture de T en c *finalise* une boucle. On va donc imposer un nombre minimal b de bases dans une boucle en remplaçant `c` par $\texttt{c}^b$ dans la définition de T. De plus, on remarque que la seule réécriture qui prolonge une échelle déjà entamée est `T -> a T b`. On impose un nombre minimal e de bases à une échelle en remplaçant les autres occurences de `a T b` par $\texttt{a}^e\ \texttt{T}\ \texttt{b}^e$. On obtient alors la grammaire de la figure 10.2.

$$\begin{array}{rcl} \texttt{ARN} & \texttt{->} & \texttt{S} \mid \varepsilon \\ \texttt{S} & \texttt{->} & \texttt{a}^e\ \texttt{T}\ \texttt{b}^e\ \texttt{S} \mid \texttt{a}^e\ \texttt{T}\ \texttt{b}^e \mid \texttt{c S} \mid \texttt{c} \\ \texttt{T} & \texttt{->} & \texttt{a}^e\ \texttt{T}\ \texttt{b}^e\ \texttt{S} \mid \texttt{a T b} \mid \texttt{c S} \mid \texttt{c}^b \end{array}$$

Fig. 10.2 : Ensemble de grammaires simples contraignant les tailles minimales b et e de boucle et d'échelles.

Cette génération est modulaire et simple à mettre en oeuvre pour quiconque est familier avec le principe de grammaire non contextuelle. Son seul inconvénient est une complexité quadratique, qui rend difficile la génération de structures d'ARNs ribosomaux 23s, dont les tailles sont autours de 3500 bases. C'est pourquoi nous avons décidé d'utiliser la génération de Boltzmann.

10.1.2 Génération uniforme selon les principes de Boltzmann

La génération de Bolzmann est particuliérement adaptée à l'usage auquel on la destine dans une démarche bioinfrmatique. En effet on peut clairement tolérer un écart autours de la taille n souhaitée dans une étude statistique, ou dans une analyse d'algorithme, surtout si n est grand, cas où la génération de Boltzmann se révèle de toute façon indispensable pour sa complexité linéaire. On trouvera le détail de la plupart des calculs en annexe.

On reprend la grammaire construite pour le cas uniforme, et on s'intéresse uniquement à son non terminal S[1]. Soit $S(z)$ sa série génératrice :

$$S(z) = z^2S(z)^2 + z^2S(z) + zS(z) + z$$

[1] On pourra revenir à un système susceptible d'engendrer ε en le choisissant avec probabilité $\frac{1}{S(x)+1}$ au début de la génération.

$$S(z) = \frac{1 - z - z^2 - \sqrt{z^4 - 2z^3 - z^2 - 2z + 1}}{2z^2}$$

On rappelle que $\rho = \frac{3+\sqrt{5}}{2}$.
$S(z)$ vérifie la condition de *moyenne* du théorème 25 :

$$\frac{\partial S(z)}{\partial z} = \frac{1}{x^2} - \frac{2}{x^3} + \frac{2 - 3x - x^2 - x^3}{x^3\sqrt{x^4 - 2x^3 - 2x + 1}}$$

Or $\lim_{x\to\rho} 2 - 3x - x^2 - x^3 = -7\sqrt{5} - 15 < 0$ et $\lim_{x\to\rho} S(z) = -\frac{4 + 2\sqrt{5}}{7 + 3\sqrt{5}} < 0$

$$\Rightarrow \lim_{x\to\rho-} x\frac{\frac{\partial S(z)}{\partial z}}{S(z)} = +\infty$$

Cependant, $S(z)$ ne vérifie pas la condition de *variance* :

$$\lim_{x\to\rho^-} \frac{\sqrt{\mathbb{E}({N_x}^2) - \mathbb{E}(N_x)^2}}{\mathbb{E}(N_x)} = i\infty$$

On trouve heureusement dans [7] une discussion sur le cas des singularités de degrés négatifs. A priori, elles sont rédibitoires, cependant on peut les contourner en *pointant* les structures combinatoires. L'opérateur de pointage isole un des atomes de la spécification.
La transposition de ce concept dans le domaines des séries génératrices va nous être utile. En effet, la S.G $S^\bullet(z)$ associée à l'équivalent pointé d'une classe combinatoire de S.G. $S(z)$ est telle que :

$$S^\bullet(z) = z\frac{\partial S(z)}{\partial z}$$

Or, la génération de structures pointées ne nécessite pas de nouvelle construction algorithmique. On utilise pour cela les règles suivantes :

$$\mathcal{A} = \mathcal{B} \times \mathcal{C} \Rightarrow \mathcal{A}^\bullet = \mathcal{B}^\bullet \times \mathcal{C} \ \cup \ \mathcal{B} \times \mathcal{C}^\bullet$$

$$\mathcal{A} = \mathcal{B} \cup \mathcal{C} \Rightarrow \mathcal{A}^\bullet = \mathcal{B}^\bullet \ \cup \ \mathcal{C}^\bullet$$

Il nous suffit donc de dupliquer les non terminaux en leur ajoutant leurs équivalent pointés.

```
 S   ->  a S b S | a S b | c S | c
S•   ->  a• S b S | a S• b S |a S b• S |a S b S•
     |   a• S b | a S• b | a S b•
     |   c• S | c S• | c•
```

Fig. 10.3 : Grammaire non contextuelle pointée satisfaisant la condition de variance.

On obtient donc avec un logiciel de calcul symbolique les probabilités à affecter à chacune réécriture pour obtenir une génération de Boltzmann sur $\mathcal{S}^\bullet$. Cette génération en temps linéaire est du à l'action de l'opérateur de pointage sur le développement asymptotique de la série génératrice. Son exposant est en effet diminué d'une unité, ce qui la fait rentrer dans une catégorie de fonctions couvertes par le théorème 5 de [7]. On en déduit immédiatement la linéarité de la génération, le nombre d'essais tendant vers une valeur dépendant de la tolérance relative ε.
Par exemple, pour $\varepsilon = 0.1$, c'est à dire 10% d'erreur tolérée sur la taille des séquences, on générera une grande séquence en ± 20 essais. Pour 1%, il faudra engendrer ± 206 séquences de tailles

n	x	a S b S	a S b	c S	c
1	0.3610240207480621	0.1215679	0.1303382	0.3610240	0.3870700
10	0.3730093873471276	0.1567408	0.1391360	0.3730093	0.3311136
100	0.3804310453876930	0.2012315	0.1447277	0.3804309	0.2736098
1000	0.3817871092438284	0.2238646	0.1457614	0.3817872	0.2485870
10000	0.3817871092438284	0.2320846	0.1458838	0.3819473	0.2400846

Fig. 10.4 : Probabilités des réécritures pour différentes valeurs de n

non admissibles. On trouvera en 10.1.2 quelques valeurs pour de x et des probabilités des différentes réécritures de S pour différentes valeurs de n. A partir de ces données, auxquelles viennent s'ajouter les probabilités des non terminaux pointés, on engendre rapidement des séquences aléatoires de grandes tailles.
On aurait aussi pu utiliser une autre approche proposée en [7] et prendre la singularité dominante ρ comme paramètre *fixe*, une génération par rejet anticipé[2] fonctionne en temps linéaire avec tolérance et en temps quadratique sans tolérance.
Notre but étant la génération aléatoire de séquences réalistes, c'est à dire le respect pour les séquences engendrées de certains paramètres statistiques observés dans les séquences réelles, on va s'intéresser à la distribution des bases dans une famille d'ARNs, les ARNs ribosomaux.

10.2 Etudes des paramètres des ARNs

Dans l'idéal, les repartitions des bases dans les séquences réelles respectent directement les comportements asymptotiques. Notre modèle étant purement structurel, ce qui militerait pour une absence d'information au niveau de la composition en bases, cela est peu crédible. Nous allons donc commencer par *scanner* une base d'ARN ribosomaux.

10.2.1 Pourquoi les ARNrs ?

Comme nous l'avons vu dans la partie biologique de ce mémoire, il y a trois types d'ARN : les ARNs messagers, les ARNs de transfert et les ARNs messagers.

Les ARNs de transfert ont une structure en trèfle qu'on peut observer dans la figure 3.9, page 23. Cette structure est très fortement contrainte seule un bourgeon y est variable. Les informations structurelles n'y sont donc pas suffisemment variées pour mériter une modélisation par une grammaire non contextuelle.

Jusqu'à assez récemment, la structure des ARN messagers n'a pas été prise en compte, car on considérait que l'ARNm étant interprété par le ribosome après avoir été étiré, c'est à dire que seules subsistent les liaisons phosphodiesters. Cependant, on commmence à tenir compte de critères structurels pour l'ARNm, car il semblerait qu'ils ne soient pas uniquement les vecteurs d'une information traduite, mais aussi acteurs dans les mécanismes d'interférence d'ARN. D'autre part, la traduction peut ne pas être déterministe, on observe par exemple des décalages de phase -1 au moment de la traduction chez certains virus eucaryotes. Cependant, cet intérêt étant assez récent, on ne dispose pas encore de suffisemment de données expérimentales.

[2] C'est à dire dès que la taille temporaire de la séquence engendrée dépasse la tolérance accordée

On étudiera donc les ARNr, dont la structure semble avoir une importance fonctionelle[3], ou du moins reflète des différences de fonctions. De plus, la quantité de structures constatées expérimentalement est élevée. Nous avons décidé de réaliser ces statistiques à partir des données de la base *The Comparative RNA Web Site* [14] sur $\pm$ 500 structures secondaires d'ARNr, soit 10Mo de données.

10.2.2 Résultats

On trouve dans la figure 10.5 un résumé rapide des statistiques observées présentant juste les compositions en sous structures. On trouvera une version plus complète de ces statistiques en annexe.
On utilisera les notations suivantes pour les sous structures : Extrémités 3' et 5'(Ex3,Ex5), Echelle (E), Renflement(R), Boucle(B), Boucle Interne(Bi), MultiBoucle(MB) On compare ces résultats

Règne:CS	NB	#Bases	%Ex3+Ex5	%E	%R	%B	%BI	%MB
Archae 5s	1	123	4,88%	61,79%	3,25%	13,82%	10,57%	4,88%
16s	18	1486	1,07%	60,06%	4,24%	11,85%	9,84%	12,87%
23s	12	2946	0,37%	56,20%	3,03%	13,25%	13,78%	13,34%
Bacteria 5s	24	120	1,76%	64,16%	2,49%	13,95%	9,32%	7,49%
16s	231	1531	1,44%	58,97%	4,35%	11,59%	10,88%	12,70%
23s	100	3325	0,23%	49,28%	2,48%	11,80%	11,43%	24,74%
Eucaryotes 5s	1	119	0,84%	62,18%	1,68%	13,45%	15,97%	5,04%
16s	38	1648	0,92%	52,93%	4,09%	12,26%	11,31%	18,43%
23s	56	2261	0,59%	50,56%	2,82%	16,94%	11,06%	18,01%

Fig. 10.5 : Pourcentages de bases appartenant aux différentes sous structures chez les ARN ribosomaux.

avec les espérances asymptotiques calculée dans la partie combinatoire du mémoire. On constate des disparités, principalement au niveau du nombre de bases présent dans des renflements (R). Cependant, ces disparités ne sont pas trop importantes, ce qui pourrait militer en faveur d'une

%Ext3+Ext5	%E	%R	%B	%BI	%MB
0%	55,27%	8,06 %	17,08%	4,98%	14,58

Fig. 10.6 : Asymptotique des pourcentages de bases dans les différentes sous structures du modèle e=1,b=1.

génération uniforme pour les structures secondaires d'ARN.

Cependant, les disparités apparaissent au niveau des tailles des sous structures. En comparant les tableaux 10.7 et 10.8. On constate des *degrès de granularité* radicalement différents, que l'on visualise très bien avec un outil comme RNAViz[1] (voir C). Une génération non uniforme des structures secondaires s'impose donc, on va donc employer la méthode récursive *amenagée* décrite par Denire, Roques et Termier dans [4].

10.3 Génération non uniforme

En réutilisant la grammaire de la partie combinatoire, on dispose d'un grand nombre de leviers pour contraindre une génération aléatoire de structure d'ARN. Cependant, les conditions d'application du théorème 22, page 82 sont loin d'être simples à vérifier sur une telle grammaire. On va donc

[3] Les décalages de phase n'arrivant par exemple pas chez tous les organismes, on peut en déduire une différence au niveau du ribosome.

Règne	CS :	NB :	#Bases	E	R	B	IL	ML	#Ins/ML
Archae	5s	1	123	9,50	1,33	8,50	6,50	6,00	3,00
	16s	18	1486	9,45	2,62	5,87	5,75	9,11	2,81
	23s	12	2946	8,16	1,93	5,59	7,53	11,79	4,05
Bacteria	5s	24	120	10,90	1,49	8,42	5,43	9,04	3,00
	16s	231	1531	8,71	2,35	5,72	5,92	8,82	2,82
	23s	100	3325	8,25	1,98	5,72	6,82	25,41	4,09
Eucaryotes	5s	1	119	12,33	2,00	8,00	9,50	6,00	3,00
	16s	38	1648	8,80	2,52	6,98	6,70	14,46	2,77
	23s	56	2261	8,21	2,12	7,31	7,71	14,44	3,19

Fig. 10.7 : Tailles moyennes des sous structures chez les ARN ribosomaux

%Ext3+Ext5	%E	%R	%B	%BI	%MB
1.236%	2	1,618	1,618	3,236	0,618

Fig. 10.8 : Asymptotique des tailles des sous structures dans le modèle e=1,b=1.

commencer par l'appliquer à un cas plus simple, dans lequel on souhaite contraindre les nombres de bases appariées.

10.3.1 Application de Drmota à un cas simple

Pour cela, on reprend encore une fois la grammaire 10.1, qu'on rappelle :

$$\begin{aligned} \texttt{ARN} &\texttt{ -> } \texttt{S} \mid \varepsilon \\ \texttt{S} &\texttt{ -> } \texttt{a S b S} \mid \texttt{a S b} \mid \texttt{c S} \mid \texttt{c} \end{aligned}$$

Ramenons nous encore une fois à la restriction de cette grammaire à S. Conformément au formalisme adopté par ce théorème, nous marquons la variable c avec une variable x.

$$\Rightarrow F(z,S,x) := z^2S(z,S,x)^2 + z^2S(z,S,x) + \pi xzS(z,S,x) + \pi xz$$

π est le poids que nous affecterons à la lettre c. Vérifions les conditions d'application du théorème 22.

1. $F(0,S,x) = 0$
2. $F(z,0,x) = \pi xz \neq 0$
3. $\frac{\partial F}{\partial S \partial S} = 2z^2 \neq 0$
4. Pour n suffisemment grand, toutes les propostions de lettres sont représenteées.
5. Il n'y a qu'un seul sommet au graphes $\Rightarrow$ il est trivialement fortement connexe.
6. Il existe une solution positive (S', z') au système d'équations :

$$\begin{cases} S(z,S,1) &= z^2S(z,S,1)^2 + z^2S(z,S,1) + \pi xzS(z,S,1) + \pi xz \\ 0 &= 1 - \frac{\partial F}{\partial S}(z,S,1) \end{cases}$$

$$S'(z) = \frac{1 - z - z^2 - \sqrt{z^4 - 2z^3\pi - 2z^2 + \pi^2z^2 - 2\pi z + 1)}}{2z^2}$$

$$z' = \frac{\pi}{2} + 1 - \frac{\sqrt{\pi^2 + 4\pi}}{2}$$

Il en existe même une deuxième, mais on les départagera plus tard.

Donc, soit (S, z) la solution du système complet, on a :

$$S(z,x) = \frac{1 - \pi xz - z^2 - \sqrt{z^4 - 2\pi xz^3\pi - 2z^2 + \pi^2x^2\pi^2z^2 - 2\pi xz + 1)}}{2z^2}$$

$$z_1(x) = \frac{\pi x}{2} + 1 + \frac{\sqrt{\pi^2x^2 + 4\pi x}}{2} \quad ou \quad z_2(x) = \frac{\pi x}{2} + 1 - \frac{\sqrt{\pi^2x^2 + 4\pi x}}{2}$$

On choisi z_2 car, d'après Drmota, $\mu_c = -\frac{\frac{\partial z}{\partial x}(1)}{z(0)}$. Or, si l'on choisi z_1, alors μ_c est négatif. On obtient donc :

$$\pi = \frac{4\mu^2}{1 - \mu^2}$$

On remarque que $0 \leq \mu < 1$. De plus, si on impose $\pi = 1$ avant d'inverser, on retrouve la proportion $\frac{1}{\sqrt{5}}$ de bases non appariées dans une structure secondaire aléatoire uniforme.

10.3.2 Grammaire complète : Discussion

Essayons maintenant d'appliquer ce théorème à notre grammaire complète. Il est alors possible de satisfaire la plupart des conditions d'application du théorème 22. Cependant, la forte connexité du graphe de dépendance de la grammaire n'est trivialement pas respectée.

On a donc cherché une grammaire équivalente, et dont le graphe de dépendance serait fortement connexe, mais cela semble impossible, à partir du moment où on souhaite distinguer les boucles. En effet, celles ci sont des élements terminaux de ma grammaire. Donc d'une part il faut un non terminal spécifique qui remplisse les boucles avec des caractères spéciaux. D'autre part, ce non terminal ne peut pas appeler un non terminal associé à une autre sous structure, donc il ne lui est pas possible de se concaténer à la composante connexe centrale. Peut être est ce une limite inhérente à une décomposition correspondant de trop près à la structure ?

Cependant, Drmota explique dans [5] la condition de forte connexité par la possible apparition de distributions *pathologiques* si la condition est violée. Or l'analyse asymptotique a prouvé que les espérances des différents caractères suivent toutes (sauf `t3`et `t5`) la loi invoquée par Drmota. Il nous faudrait donc des études plus poussées dans le domaine de la combinatoire analytique pour déterminer si ces conditions sont nécessaires. De plus, les résultats du théorème de Drmota vont plus loin que donner une formule pour la proportion μ des symboles à l'asymptotique. Il en déduit la variance et une expansion des coefficients à $O(\sqrt{n})$ près.

Il n'est donc pas encore possible d'appliquer ces principes à la grammaire complète, cependant nous pensons prouver à court terme d'adaptation de la grammaire au cadre du théorème.

BIBLIOGRAPHIE

[1] Peter De Rijk an Jan Wuyts and Rupert De Wachter. Rnaviz2: an improved representation of rna secondary structure. *Bioinformatics*, 2(19):299–300, 2003.

[2] M. Vauchaussade de Chaumont and G. Viennot. Polynômes orthogonaux et problèmes d'énumération en biologie moléculaire. *Séminaire Lotharingien de Combinatoire*, 1983.

[3] A. Denise. Structures aléatoires, modèles et analyse des génomes, 2001. *Mémoire d'habilitation à diriger des recherches*.

[4] A. Denise, O. Roques, and M. Termier. Random generation of words of context-free languages according to the frequencies of letters. In D. Gardy and A. Mokkadem, editors, *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and probabilities*, Trends in Mathematics, pages 113–125. Birkhaüser, 2000.

[5] Michael Drmota. Systems of functional equations. *Random Structures and Algorithms*, 10(1-2):103–124, 1997.

[6] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Random sampling from Boltzmann principles. In P. Widmayer et al., editor, *Automata, Languages, and Programming*, number 2380 in Lecture Notes in Computer Science, pages 501–513. Springer Verlag, 2002. Proceedings of the 29th ICALP Conference, Malaga, July 2002.

[7] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probablity, and Computing*, 2003. A paraitre.

[8] L.Noe et G.Kucherov. Communication personnelle.

[9] Philippe Flajolet, Bruno Salvy, and Paul Zimmermann. Automatic average-case analysis of algorithm. *Theoretical Computer Science*, 79(1):37–109, 1991.

[10] Philippe Flajolet, P. Zimmermann, and B. Van Custem. A calculus for the random generation of combinatorial structures. Technical Report RR-1830, INRIA, 1993.

[11] F.Lefebvre. A grammar-based unification of several alignment and folding algorithms. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 143–154. AAAI Press, 1996.

[12] Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler. Combinatorics of RNA secondary structures. *Discr. Appl. Math.*, 88:207–237, 1998.

[13] Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler. Combinatorics of rna secondary structures. *Discrete Applied Mathematics*, 88:207–237, 1998.

[14] Cannone J.J., Subramanian S.and Schnare M.N.and Collett J.R.and D'Souza L.M.and Du Y.and Feng B.and Lin N.and Madabusi L.V.and Muller K.M.and Pande N.and Shang Z.and Yu N., and Gutell R.R. The comparative rna web (crw) site: An online database of comparative sequence and structure information for ribosomal, intron, and other rnas. *BioMed Central Bioinformatics*, 3(2), 2002.

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

[16] M. Nebel. Combinatorial properties of rna secondary structures, 2001. Frankfurter Informatik-Berichte 2/01, Institut für Informatik, Johann Wolfgang Goethe-Universität, Frankfurt a. M., 2001.

[17] M. Nebel. A unified approach to the analysis of horton-strahler parameters of binary tree structures, 2001. Frankfurter Informatik-Berichte 1/01, Institut für Informatik, Johann Wolfgang Goethe-Universität, Frankfurt a. M., 2001.37.

[18] P. Nicodeme, B. Salvy, and P. Flajolet. Motif statistics, 1999.

[19] M. Regnier. Generating functions in computational biology, 1997. *Lecture at Algorithms Seminar, INRIA*.

[20] D.B. Searls. Formal language theory and biological macromolecules. *Series in Discrete Mathematics and Theoretical Computer Science*, 47:117–140, 1999.

[21] F. Tahi, S. Engelen, and M. Régnier. A fast algorithm for RNA secondary structure prediction including pseudoknots approach, 2003. to be presented at BIBE'03, Washington DC.

[22] M. Vauchaussade de Chaumont and X.G. Viennot. Enumeration of RNA's secondary structures by complexity. In V. Capasso, E. Grosso, and S.L. Paven-Fontana, editors, *Mathematics in Medecine and Biology*, volume 57 of *Lecture Notes in Biomathematics*, pages 360–365, 1985.

[23] J. Waldispühl, B. Behzadi, and J.-M. Steyaert. An approximate matching algorithm for finding (sub-)optimal sequences in s-attributed grammars. In *Proceedings of the first European Conference on Computational Biology, ECCB 2002*, volume 18 of *Bioinformatics*, pages 250–259. OXFORD University Press, 2002.

[24] M. S. Waterman. Secondary structure of single stranded nucleic acids. *Advances in Mathematics Supplementary Studies*, 1(1):167–212, 1978.

[25] J.D. Watson and F. H. C. Crick. Molecular structures of nucleic acids. *Nature*, 4356:737–738, 1953.

[26] Herbert S. Wilf. A unified setting for sequencing, ranking, andselection algorithms for combinatorial objects. *Advances in Mathematics*, 24:281–291, 1977.

[27] Herbert S. Wilf. *generatingfunctionology*. Academic Press Inc., Boston, second edition, 1994.

[28] M. Zucker. Rna folding by energy minimization.

Définitions

ANNEXE

A. GÉNÉRATION ALÉATOIRE DE CHEMINS CULMINANTS

A.1 Introduction

Le problème de la génération aléatoire de chemins culminants nous a été présenté par G. Kucherov et L.Noe qui, au cours de travaux sur des algorithmes d'alignement global de sequences, ont été amenés à engendrer des alignements culminants aléatoires sans insertions/deletions. Un alignement sans indels peut être vu comme une séquences de bits 0 ou 1, où 0 représente un *mismatch* et 1 un *match*. Chaque *match* crédite l'alignement de a points, et chaque *mismatch* débite l'alignement de b points. Un alignement est alors dit culminant si son score total est supérieur à ceux de ses sous-parties.

A.2 Les chemins culminants

On substitue aux couples séquence/score des chemins positifs culminants à pas a-montants et b-descendants.

Définition 24 (Chemin culminant) :
Un chemin C positif culminant à pas a-montants et b-descendants est un chemin de $\mathcal{Z}^2$ formé de pas $(1, a)$ et $(1, -b)$ allant de $(0, 0)$ à (n, h) tel que :

- *Le chemin C est positif :* $\forall (k, y) \in C, y > 0$
- *Le chemin C culmine en n :* $\forall k < n, (k, y) \in C \Rightarrow y < h$

On notera C_n l'ensemble des chemins positifs culminants.
On remarquera que la condition de positivité traduit l'optimalité de l'alignement. En effet, un chemin ayant un point d'ordonnée négative est susceptible d'être optimisé par son suffixe démarrant en son point d'ordonnée la plus faible.

A.3 Génération uniforme de chemins culminants

Une approche classique pour la génération aléatoire d'objets décomposables consiste à compter les nombres d'objets n_X accessibles après un choix X, pour toutes les valeurs possibles de X. Si les ensembles d'objets potentiellements construits sont disjoints pour toutes les valeurs de X, alors on en déduit les probabilités de choisir les différentes valeurs de X afin de respecter l'uniformité des structures générées.
On préfère une telle approche à une génération par rejet, dont la complexité dépend trop fortement des paramètres a et b.

A.3.1 Technique classique de génération uniforme séquentielle

Soit L un ensemble de mots sur l'alphabet $X = \{x_1, x_2, \ldots, x_m\}$. Soit L_n la restriction de cet ensemble aux mots de taille n.
Soit w_p le préfixe d'un mot de L_n, on appelle $p_x(w_p)$ la probabilité que w_p soit suivi par $x \in X$ dans un mot de L_n.

$$p_x(w_p) = \frac{|\{w' | w_p.x.w' \in L_n\}|}{|\{w' | w_p.w' \in L_n\}|}$$

Proposition 2 :
Si l'on sait calculer les $p_x(w_p)$, alors on sait engendrer uniformément des mots de L_n.

Preuve : *En effet, partant du préfixe ϵ, on le prolonge par des lettres c_1, c_2, c_3, $\ldots$, c_n choisies selon les probabilités $p_{c_1}(\epsilon)$, $p_{c_2}(c_1)$, $p_{c_3}(c_1.c_2)$, $\ldots$, $p_{c_n}(c_1.c_2.\cdots.c_{n-1})$. La probabilité d'émission d'un mot $w = c_1.c_2.c_3.\cdots.c_n$ est donc égale à :*

$$\begin{aligned} p(c_1....c_n) &= p_{c_1}(\epsilon) * p_{c_2}(c_1) * p_{c_3}(c_1.c_2) * \cdots * p_{c_n}(c_1.c_2.\cdots.c_{n-1}) \\ &= \frac{|\{w'|c_1.w' \in L_n\}|}{|L_n|} * \frac{|\{w'|c_1.c_2.w' \in L_n\}|}{|\{w'|c_1.w' \in L_n\}|} * \ldots * \frac{|\{w'|c_1,\ldots,c_n.w' \in L_n\}|}{|\{w'|c_1,\cdots,c_{n-1}.w' \in L_n\}|} \\ &= \frac{|\{w'|c_1.w' \in L_n\}|}{|\{w'|c_1.w' \in L_n\}|} * \frac{|\{w'|c_1.c_2.w' \in L_n\}|}{|\{w'|c_1.c_2.w' \in L_n\}|} * \ldots * \frac{|\{w'|c_1,\ldots,c_n.w' \in L_n\}|}{|L_n|} \\ &= \frac{1}{|L_n|} \text{ car } \{w'|c_1,\ldots,c_n.w' \in L_n\} = \epsilon \end{aligned}$$

Tous les mots de L_n étant choisi par ce processus avec une probabilité $\frac{1}{|L_n|}$, ce procédé de génération aléatoire est donc uniforme.

Touver un algorithme de génération aléatoire uniforme de mots de ce langage revient donc à compter les suffixes des mots du langage étudié.

A.3.2 Application aux chemins culminants

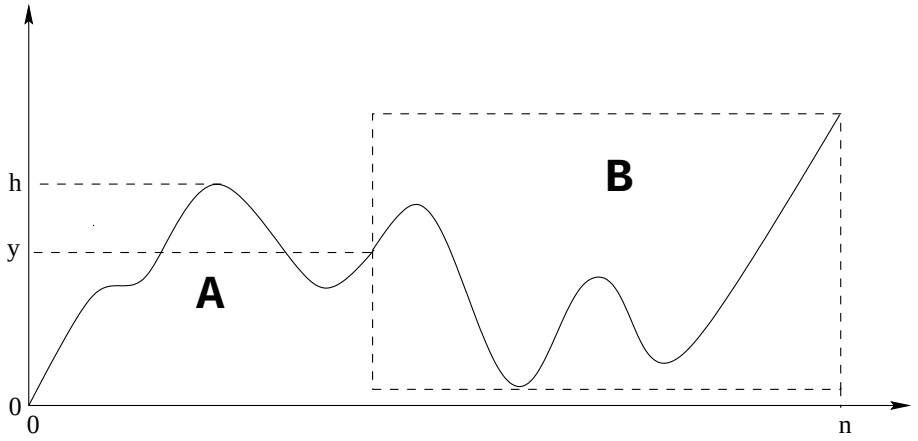


Fig. A.1 : Les suffixes B associés à un préfixe A de C_n dépendent uniquement des ordonnées h maximale atteinte et y actuelle

Dans le cas général, on voit bien qu'il est déraisonnable de précalculer les $p_x(w_p)$, m^n valeurs devant être stockées. Cependant, dans le cas des chemins culminants, il n'est pas nécessaire de compter les différents suffixes pour tous les mots sur X^n. Au moment de prolonger un préfixe de C_n, les seules données d'intérêt sont l'ordonnée maximale h atteinte par le chemin au sein du préfixe et l'ordonnée y à laquelle arrive le préfixe (voir Figure A.1). On introduit donc la notion de chemin culminant (h,y)-initié.

Définition 25 (Chemin culminant (h,y)-initié) :
Un chemin culminant (h,y)-initié est un chemin positif culminant formé de pas $(1, a)$ et $(1, -b)$ allant de $(0, y)$ à (n, y'), $y' > h$

Soit $C_{y,h,n}$ l'ensemble des chemins culminants (h,y)-initiés de taille n.

Proposition 3 :
$\forall w_p$, *chemin de $(0, 0)$ à (k, y) de hauteur max. atteinte h :*
$\{w'|w_p.w' \in C_n\} = C_{y,h,n}$

On dénombre alors les chemins culminants (h,y)-initiés de taille n pour toutes les valeurs possibles des paramètres h, y et n, et on en déduit immédiatement les valeurs des $p_x(w_p)$. Pour cela, on exhibe une décomposition récursive des chemins culminants (h,y)-initiés :

Proposition 4 :

$$C_{y,h,n} = \begin{cases} \{1\}.C_{y+a,Max(h,y+a),n-1} \bigcup \{0\}.C_{y-b,h,n-1} & \textit{Si } y > b \\ \{1\}.C_{y+a,Max(h,y+a),n-1} & \textit{Sinon} \end{cases}$$

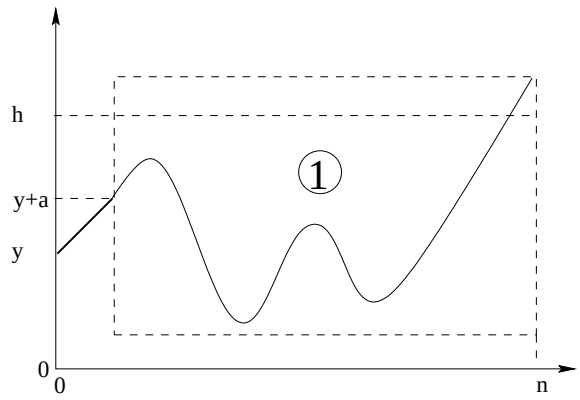
$$C_{y,h,1} = \begin{cases} \emptyset & \textit{Si (y+a} \leq \textit{h) ou (y} < \textit{0)} \\ \{1\} & \textit{Sinon} \end{cases}$$

Preuve : *Commençons par les chemins de* $C_{y,h,1}$ *:*

- *Un chemin de taille* 1 *ne peut être un pas* $(1, -b)$ *(0) car alors l'ordonnée du point d'absisse* 0 *est supérieure à celle du point d'absisse* 1*, ce qui viole la condition de chemin culminant.*
- *Un chemin constitué d'un unique pas ascendant ne peut être considéré comme un chemin culminant (h,y)-initié que si il respecte les contraintes de positivité (*$y >= 0$*) et finit en une ordonnée supérieure à* h *(*$y + a > h$*).*

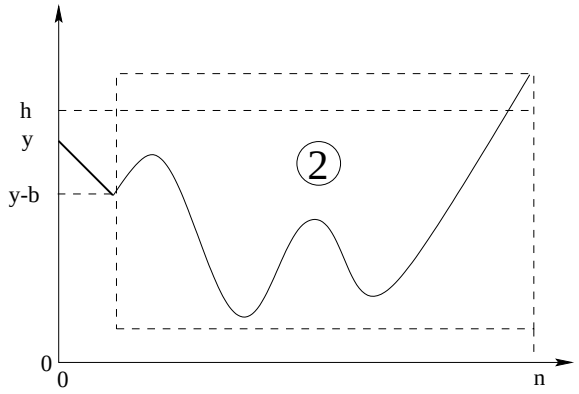
Prenons ensuite un chemin Γ *de* $C_{y,h,n}, n > 1$ *:*

- Γ *commence par un pas montant* 1 *:*



  Γ *est culminant, son ordonnée d'arrivée est donc à la fois supérieure à* h *et à* $y + a$*.*
  $\Rightarrow \mathbf{1}$*. est un chemin reliant* $(1, y + a)$ *à* (n, y')*,* $y' > h$ *et* $y' > y + a$*.*
  De plus, Γ *est positif, donc* $\mathbf{1}$*. l'est aussi. Enfin,* $\mathbf{1}$*. est culminant. En effet, si* $\mathbf{1}$*. n'est pas culminant, alors* Γ *ne l'est pas non plus.*
  $\Rightarrow \mathbf{1}$*. est un chemin de* $C_{y+a,Max(h,y+a),n-1}$

- Γ *commence par un pas descendant* 0 *:*



On remarque que $y > b$. En effet, si $y \leq b$, alors le pas descendant viole la contrainte de positivité stricte. Γ est culminant, son ordonnée d'arrivée est cette fois supérieure à h.
*$\Rightarrow$ **2**. est un chemin reliant $(1, y-b)$ à (n, y'), $y' > h$*
*De plus, Γ est positif, donc **2**. l'est aussi. Enfin, de même que pour **1**., **2**. est culminant.*
*$\Rightarrow$ **2**. est un chemin de $C_{y-b,h,n-1}$ si $y > b$.*

On remarquera que la décomposition récursive exhibée n'admet aucune ambiguïté, car les objets énumérés commencent par des caractères différents pour chacunes des opérandes de l'union. On peut donc transposer cette relation sur des ensembles sur les cardinaux de ceux ci, ce qui nous permet de proposer l'algorithme suivant.

A.3.3 Algorithme

A partir de cette récurrence, on calcule les nombres $C(y, h, n)$ de chemins de taille N, pour toutes les valeurs des paramètres y et h.

Algorithme (Comptage):

```
Pour tout h∈[0,a*N] et y∈[0,a*N]
   Si (y+a>h)
   Alors C(y,h,1) := 1;
   Sinon C(y,h,1) := 0;

Pour n allant de 2 à N
Pour tout h∈[0,a*N] et y∈[0,a*N]
   Si (y>b)
   Alors C(y,h,n) := C(y+a,Max(h,y+a),n-1) + C(y-b,h,n-1);
   Sinon C(y,h,n) := C(y+a,Max(h,y+a),n-1);
```

Une fois cette table remplie, on sait selon quelles probabilités prolonger un chemin positif d'un pas a-ascendant ou d'un pas b-descendant.

Algorithme (Génération):

```
y := 0;
h := 0;
S := ϵ;
Pour n allant de N à 2
```

```
   p_{x,S} := C(y+a,Max(h,y+a),n-1)/C(y,Max(h),n);
   Si (ran()<p_{x,S})
   Alors y := y+a; h := max(h,y); S := S + '1';
   Sinon y := y-b; S := S + '0';
S := S + '1';
```

A.3.4 Complexités

La phase de comptage a une complexité en temps et en espace de $O(N^3 * a^2)$.
La complexité en temps est ici donnée en opérations arithmétiques. Cependant, les nombres manipulés croissant exponentiellement avec N, on doit intégrer à la complexité de l'algorithme le surcoût engendré par l'emploi d'arithmétique en précision arbitraire. Ce surcoût étant de l'ordre, pour chaque opération, de la somme des logarithmes des opérandes, on peut conjoncturer un comportement en $O(N^4 * a^2)$ de la complexité *machine*.
La phase de génération est quant à elle linéraire en opérations arithmétiques et en espace. De même, on peut conjecturer un comportement en $O(N^2)$ lorsque les nombres manipulés croissent exponentiellement sur N.

Enfin, si l'on ne souhaite engendrer que les chemins culminants de hauteur donnée h, on peut alors obtenir un algorithme de complexité $O(N * h)$ en modifiant très légèrement la récurrence permettant le comptage des coefficients.

$$C_{y,n} = \begin{cases} \{1\}.C_{y+a,n-1} & \text{Si } y \leq b \\ \{0\}.C_{y-b,n-1} & \text{Si } y + a \geq h \\ \{1\}.C_{y+a,n-1} \bigcup \{0\}.C_{y-b,n-1} & \text{Sinon} \end{cases}$$

$$C_{y,1} = \begin{cases} \{1\} & \text{Si } (y + a = h) \\ \emptyset & \text{Sinon} \end{cases}$$

B. GENRGENS : GENERATION OF RANDOM GENOMIC SEQUENCES

B.1 Contexte

Un des problèmes importants liés à l'analyse *in silico* des génomes génomiques est l'évaluation du "bruit de fond". Il s'agit de déterminer dans quelle mesure des structures ou des propriétés observées sur des séquences biologiques déduites ont une réelle signification ou sont là "par hasard". Le raisonnement qui prévaut est le suivant : si une propriété est réellement signifiante d'un point de vue biologique, alors elle ne doit pas se manifester dans une séquence aléatoire, ou s'y manifester de façon très différente par rapport à une séquence biologique. Dans certains cas, comme l'évaluation de la sur-représentation de motifs, il existe des méthodes purement analytiques pour comparer séquences biologiques et séquences aléatoires. Dans d'autres cas, il est nécessaire de recourir à la simulation, c'est-à-dire la génération aléatoire *in silico* de séquences. Il s'agit d'engendrer des séquences de longueur n fixée, obéissant à un modèle déterminé. Il est donc intéressant pour un biologiste voulant valider une hypothèse de pouvoir engendrer des séquences aléatoires respectant des contraintes. D'autre part, des séquences aléatoires permettent la calibration, voir la validation d'outils d'inférence. Face à cette demande, nous avons conçu un logiciel mettant en oeuvre différentes techniques de générations aléatoires, GenRGenS.

B.2 GenRGenS

GenRGenS (Generation of Random Genomic Sequences) est une boîte à outils pour la génération aléatoire de séquences selon différents types de modèles. D'un usage simple (Interface Graphique) et développé en Java, ce qui le rend exécutable sur toutes les plateformes, il est destiné à des bioinformaticiens sensibilisés aux formalismes linguistiques et/ou statistiques. GenRGenS est téléchargeable avec ses sources dans sa version 1.0 à l'adresse :

`www.lri.fr\ ~denise\GenRGenS`

Les modèles de génération actuellement supportés par GenRGenS sont le modèle markovien et le modèle des grammaires non contextuelles pondérées.

B.2.1 Modèle markovien

Ce modèle bien connu impose des contraintes statistiques aux occurrences de sous-séquences dans les séquences engendrées. Il est paramétré par un entier, l'orde de la chaîne de Markov, qui détermine la taille des sous-séquences considérées. GenRGenS permet de gérer des chaînes de Markov hétérogènes à plusieurs phases (cas des séquences codant pour l'ARN) et certains modèles de Markov cachés. En ce qui concerne la génération, [1] on dispose d'un algorithme linéaire sur la taille de la séquence voulue, qui traite aussi les variantes hétérogènes. La figure 1 présente un exemple de *fichier de description* qui est fourni en entrée à GenRGenS pour engendrer des séquences aléatoires markoviennes. Un tel fichier de description peut être réalisé automatiquement par GenRGenS si on lui fournit une séquence de référence et certains paramètres comme l'ordre de la chaîne et le nombre de phases. De plus, l'approche markovienne présente l'avantage d'être entièrement automatisable quand la linguistiques markovienne est déduite du séquencement d'un matériel génétique réél. Un petit utilitaire fourni avec GenRGenS et piloté par son interface graphique construit alors automatiquement le fichier de description markovien associé.

B.2.2 Génération avec contraintes syntaxiques

Moins populaire auprès des biologistes que le modèle markovien, ce type de génération est basé sur le formalisme des grammaires non contextuelles (ou algébriques). Il engendre des mots pour

[1] W.M. Fitch. Random Sequences. Journal of Molecular Biology, 163:171-176, 1983

```
TYPE = MARKOV
ORDER = 2
START =
  ggg 7
  cca 3
FREQUENCIES =
  aaa 00 aca 50 aga 12 ata 56       caa 23 cca 86 cga 10 cta 10
  aac 00 acc 78 agc 12 atc 23       cac 15 ccc 78 cgc 21 ctc 51
  aag 00 acg 51 agg 51 atg 23       cag 46 ccg 46 cgg 32 ctg 35
  aat 00 act 32 agt 53 att 02       cat 91 cct 45 cgt 51 ctt 11

  gaa 21 gca 84 gga 51 gta 31       taa 86 tca 84 tga 02 tta 61
  gac 12 gcc 64 ggc 51 gtc 51       tac 86 tcc 43 tgc 23 ttc 31
  gag 51 gcg 61 ggg 35 gtg 18       tag 48 tcg 48 tgg 35 ttg 18
  gat 86 gct 43 ggt 84 gtt 68       tat 65 tct 56 tgt 81 ttt 81
```

Fig. B.1 : Un fichier de description GenRGenS pour un modèle markovien d'ordre deux. La clause START permet de déterminer une loi pour le début de la séquence : elle commencera par ggg 7 fois sur 10, par cca 3 fois sur 10. Dans la clause FREQUENCIES, le nombre qui suit chaque trinucléotide est le nombre d'occurrences observées dans une séquence biologique de référence. Les probabilités associées au modèle sont calculées en conséquence par GenRGenS.

lesquels les contraintes ne sont plus uniquement statistiques, mais aussi structurelles. Les grammaires formelles permettent de décrire des séqeunces soumises à des contraintes structurelles, comme par exemple la présence ou l'absence de motifs particuliers dans les séquences, ou des interactions à distance dans les séquences d'ARN. Ainsi, le codage classique des structures secondaires par des *mots de Motzkin* [22, 20], selon lequel un nucléotide non apparié est représenté par une lettre c et les deux nucléotides $5'$ et $3'$ d'un appariement sont respectivement codés par un a et un b, définit un langage qui peut être décrit et engendré par une grammaire non contextuelle (figure 2). Le concept de *grammaire pondérée*, présenté dans [4], permet d'engendrer des séquences ainsi structurées tout en respectant des contraintes statistiques données portant sur les fréquences des lettres. A chaque lettre est attribuée un *poids* qui influe sur son nombre moyen d'occurrences dans les séquences engendrées. L'algorithme de génération [4] est linéaire en temps sur la taille des mots souhaités après un précalcul de complexité quadratique. Le principe de génération offre aussi la possibilité d'adjoindre aux contraintres syntaxiques des contraintes statistique portant sur la représentation des symboles terminaux, ce au moyen d'un système de poids. La génération avec contraintes grammaticales est donc non seulement paramétrée par une grammaire, mais aussi par un ensemble de couples *Symboles Terminaux/Poids*.

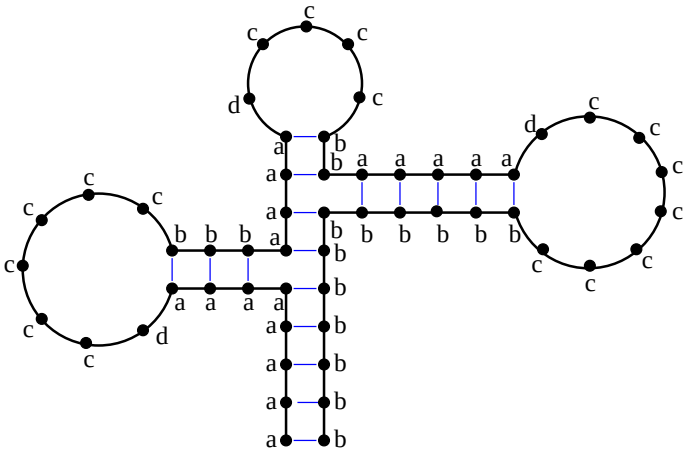
B.3 Perspectives

Bientôt, GenRGenS s'enrichira d'autres modes de génération :

1. Un mode "chainé" qui permettra de cumuler différents niveaux d'abstractions. Par exemple, la structure d'une séquence pourra être définie par une grammaire de "haut niveau" quand son détail sera défini par des liguistiques markoviennes.

2. Des modes permettant d'engendrer des séquences aléatoires à partir d'expressions régulières ou PROSITE, de profils HMM et de profils généralisés.

D'autre part, des techniques de génération aléatoires basées sur des formalismes capables de modéliser des pseudo-noeuds sont à l'étude. Les grammar non-contextuelles sont en effet d'expressivités



```
TYPE = GRAMMAR
SYMBOLS = LETTERS
RULES
   S -> a S b S;
   S -> a S b;
   S -> d T c;
   T -> c T;
   T -> c c c;
WEIGHTS =
   a 16
   c 2
   d 1
```

Fig. B.2 : Une structure secondaire qui correspond au mot `aaaaaaaadccccccbbbaaaadccccb-baaaaadccccccbbbbbbbbbbbb`, et une grammaire pondérée du langage. Au codage classique on a ajouté la lettre `d` qui permet, par le biais des pondérations, de faire varier le nombre moyen de boucles dans les structures engendrées aléatoirement. Plus le poids d'une lettre est élevé, plus sa fréquence est élevée dans les mots engendrés. Ici, la forte pondération de `a` indique que l'on désire favoriser les longueurs des tiges par rapport à celles des boucles.

insuffisante pour modéliser de telles structures. Un des formalismes étudié actuellement est le formalisme de String Variable Grammars de Searl qui feront l'objet d'un ajout à GenRGenS si une procédure de génération aléatoire de complexité raisonnable est trouvée.

C. COMPARAISON GÉNÉRATION UNIFORME, SÉQUENCE RÉELLE

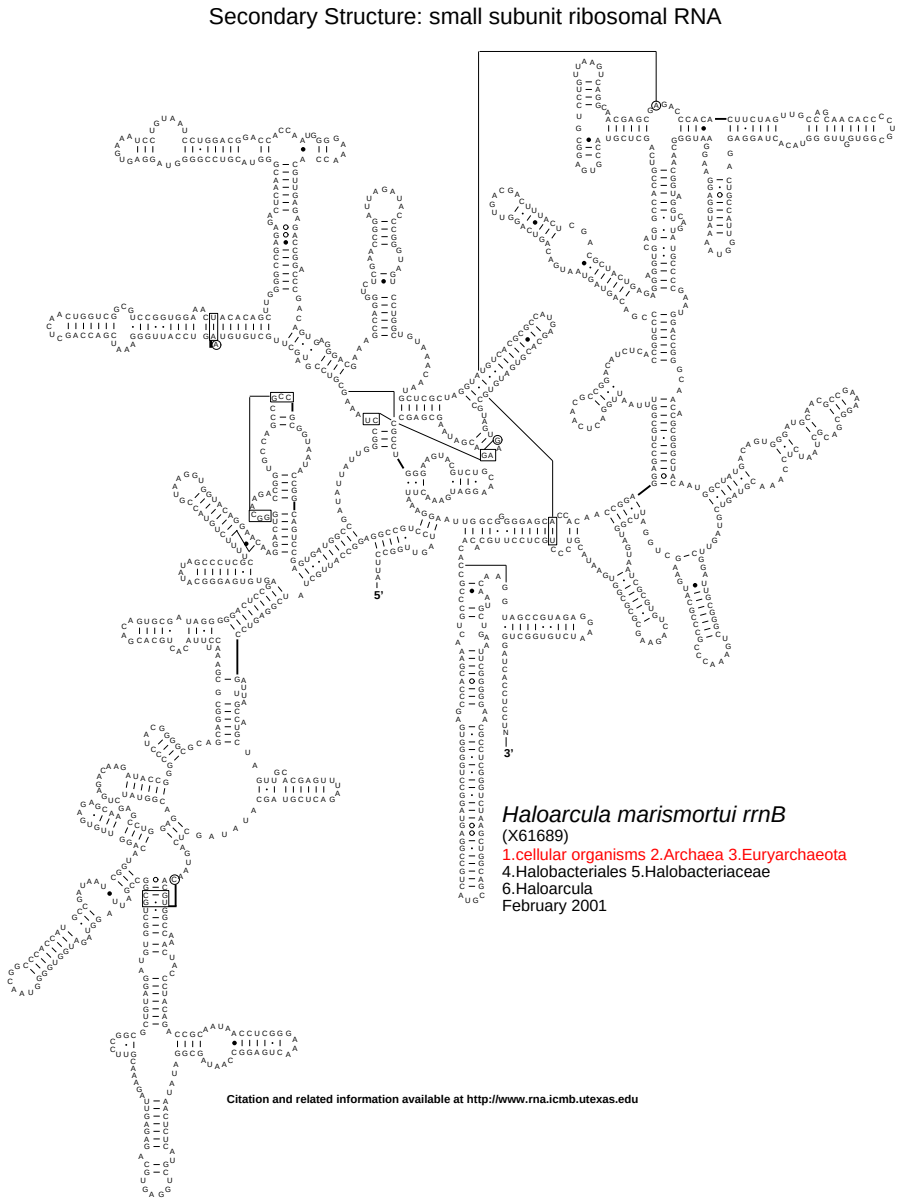


Fig. C.1 : Structures secondaires réelle d'ARNr 16*s*

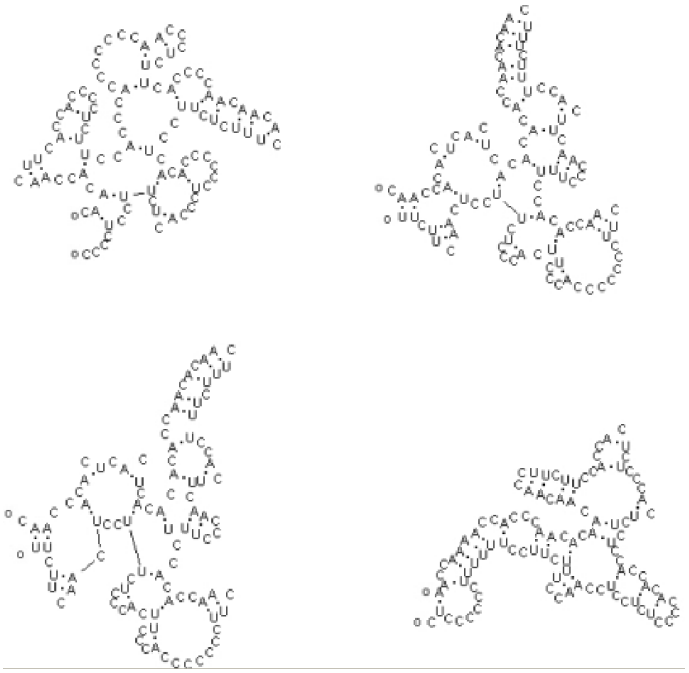


Fig. C.2 : Structures secondaires aléatoires uniformes

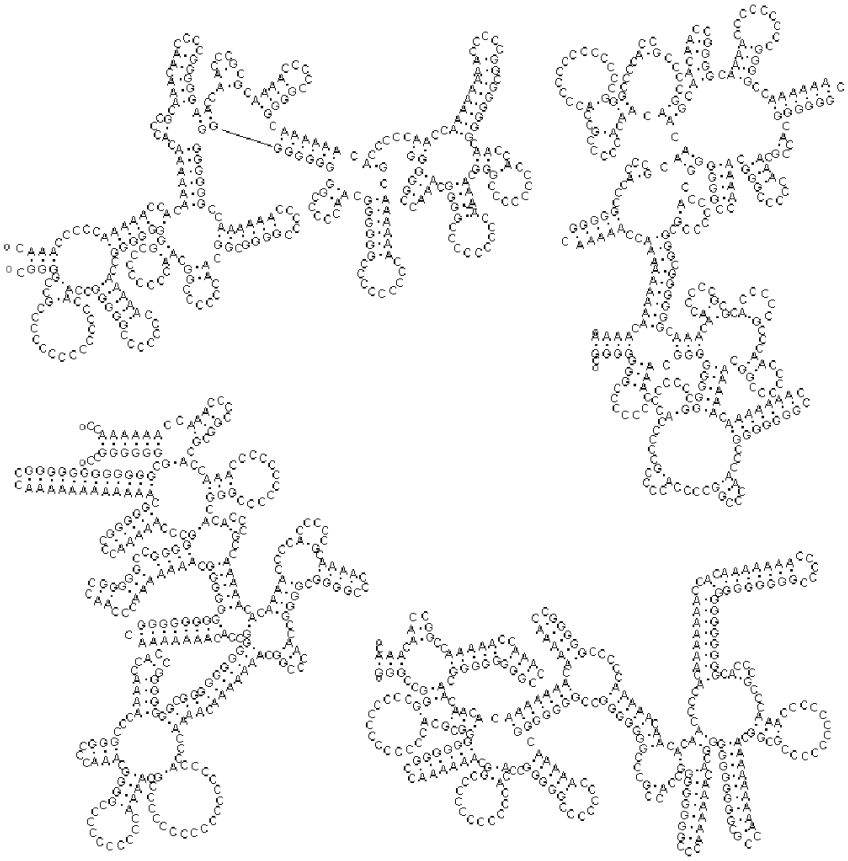


Fig. C.3 : Structures secondaires aléatoires pondérées

D. STATISTIQUES STRUCTURELLES BRUTES DES ARNR

Règne	CS	NB	#B	#B PK	5'	3'	#E	#B/E	#R	#B/R	#B	#B/B	#BI	#B/BI	#BI	#B/BI	#Remp	#Remp/MB
Archae	5s	1	123	0	3	3	8	76	3	4	2	17	2	13	1	6	3	3
	16s	18	1486,06	37,56	3,67	12,17	94,5	892,56	24,06	63,06	30	176,06	25,44	146,28	21	191,28	59	2,81
	23s	12	2946,67	115,67	1,67	9,17	203,83	1656	46,17	89,17	69,83	390,33	54	406,17	33,83	393,17	136,5	4,05
Bacteria	5s	24	120,67	0	0,83	1,29	7,13	77,42	2,04	3	2	16,83	2,08	11,25	1	9,04	3	3
	16s	231	1531,06	36,2	8,42	13,6	103,73	902,88	28,43	66,55	31,07	177,52	28,16	166,58	22,07	194,51	62,21	2,82
	23s	100	3325,42	334,3	2,14	5,64	198,66	1638,88	41,76	82,42	68,66	392,48	55,72	380,18	32,52	822,68	132,76	4,09
Eucaryotes	5s	1	119	0	0	1	6	74	1	2	2	16	2	19	1	6	3	3
	16s	38	1648,5	32,34	5,61	9,61	98,55	872,53	27,08	67,45	29,18	202,03	27,84	186,5	20,45	303,79	57,08	2,77
	23s	56	2261,61	31,39	7,93	5,39	135,79	1143,43	13,62	63,86	51,39	383,14	32,18	250,04	28,29	407,36	92,54	3,19

Légende

#B/X = Nombre de bases comprises dans des sous structures X
#X = Nombre total de sous structures X
#B = Taille moyenne de la séquence
#B PK = Nombre moyen de bases impliquées dans des pseudonoeuds
#Remp/MB = Nombre moyen de remplissages par multiboucles

Où **X** a pour valeur :

E = Echelle **R** = Remplissage **B** = Boucle **BI** = Boucle Interne **MB** = MultiBoucle

Fig. D.1 : Statistique des sous structures

E. APPLICATION SIMPLE DU THÉORÈME DE DRMOTA

```
> restart;
```

Drmota "simple"

On utilise la restriction de la grammaire suivante à S, qui est une formulation alternative de la grammaire des structures secondaires d'ARN et on lui applique le résultat de Drmota sur les systèmes d'équations :

$$\mathrm{A} ::= \mathrm{S} \mid \varepsilon$$

$$\mathrm{S} ::= \mathrm{a\,S\,b\,S} \mid \mathrm{a\,S\,b} \mid \mathrm{c\,S} \mid \mathrm{c}$$

```
> F:=(z,S,x)-> z^2*S^2 + z^2*S + pi*x*z*S + pi*x*z;
```

$$F := (z,\, S,\, x) \to z^2\, S^2 + z^2\, S + \pi\, x\, z\, S + \pi\, x\, z$$

On a bien $\mathrm{F}(0,\, S,\, x) = 0$ et $\mathrm{F}(z,\, 0,\, x) \neq 0$.

De plus, la grammaire est fortement connexe, le système d'équations n'est pas linéaire, et la condition des cônes est respectée.

On doit cependant encore vérifier qu'il existe des solutions positives S_0 et z_0 à :

$$S = \mathrm{F}(z,\, S,\, 1)$$

$$0 = 1 - (\frac{dF}{dS})(z,\, S,\, 1)$$

```
> FS:=unapply(diff(F(z,S,x),S),z,S,x);
```

$$FS := (z,\, S,\, x) \to 2\, z^2\, S + z^2 + \pi\, x\, z$$

```
> eq1:=S=F(z,S,1);
```

$$eq1 := S = z^2\, S^2 + z^2\, S + \pi\, z\, S + \pi\, z$$

```
> eq2:=0=1-FS(z,S,1);
```

$$eq2 := 0 = 1 - 2\, z^2\, S - z^2 - \pi\, z$$

```
> solve(eq1,S);
```

$$\frac{-z^2 - \pi\, z + 1 + \sqrt{z^4 - 2\, z^3\, \pi - 2\, z^2 + \pi^2\, z^2 - 2\, \pi\, z + 1}}{2\, z^2},$$
$$\frac{-z^2 - \pi\, z + 1 - \sqrt{z^4 - 2\, z^3\, \pi - 2\, z^2 + \pi^2\, z^2 - 2\, \pi\, z + 1}}{2\, z^2}$$

On choisit la solution positive pour S_0.

```
> S[0] := %[2];
```

$$S_0 := \frac{-z^2 - \pi\, z + 1 - \sqrt{z^4 - 2\, z^3\, \pi - 2\, z^2 + \pi^2\, z^2 - 2\, \pi\, z + 1}}{2\, z^2}$$

```
> sol3 := solve(subs(S = S[0],eq2),z);
```

$$sol3 := \frac{\pi}{2} - 1 + \frac{\sqrt{\pi^2 - 4\,\pi}}{2},\ \frac{\pi}{2} - 1 - \frac{\sqrt{\pi^2 - 4\,\pi}}{2},\ \frac{\pi}{2} + 1 + \frac{\sqrt{\pi^2 + 4\,\pi}}{2},\ \frac{\pi}{2} + 1 - \frac{\sqrt{\pi^2 + 4\,\pi}}{2}$$

```
> z[0][1] := sol3[1];
```

$$z_{01} := \frac{\pi}{2} - 1 + \frac{\sqrt{\pi^2 - 4\,\pi}}{2}$$

```
> z[0][2] := sol3[2];
```

$$z_{02} := \frac{\pi}{2} - 1 - \frac{\sqrt{\pi^2 - 4\,\pi}}{2}$$

```
> z[0][3] := sol3[3];
```

$$z_{03} := \frac{\pi}{2} + 1 + \frac{\sqrt{\pi^2 + 4\,\pi}}{2}$$

```
> z[0][4] := sol3[4];
```

$$z_{04} := \frac{\pi}{2} + 1 - \frac{\sqrt{\pi^2 + 4\,\pi}}{2}$$

Il existe une solution positive (z_0,S_0), z_0 restant à choisir ultérieurement entre z_{03} et z_{04} solutions réelles positives.

On a vérifié les conditions d'applications du théorème, on passe maintenant à son application :

```
> eq3:=S=F(z,S,x);
```

$$eq3 := S = z^2\,S^2 + z^2\,S + \pi\,x\,z\,S + \pi\,x\,z$$

```
> eq4:=0=1-FS(z,S,x);
```

$$eq4 := 0 = 1 - 2\,z^2\,S - z^2 - \pi\,x\,z$$

```
> solve(eq3,S);
```

$$\frac{-z^2 - \pi\,x\,z + 1 + \sqrt{z^4 - 2\,z^3\,\pi\,x - 2\,z^2 + \pi^2\,x^2\,z^2 - 2\,\pi\,x\,z + 1}}{2\,z^2},$$
$$\frac{-z^2 - \pi\,x\,z + 1 - \sqrt{z^4 - 2\,z^3\,\pi\,x - 2\,z^2 + \pi^2\,x^2\,z^2 - 2\,\pi\,x\,z + 1}}{2\,z^2}$$

```
> S[1] := %[2];
```

$$S_1 := \frac{-z^2 - \pi\,x\,z + 1 - \sqrt{z^4 - 2\,z^3\,\pi\,x - 2\,z^2 + \pi^2\,x^2\,z^2 - 2\,\pi\,x\,z + 1}}{2\,z^2}$$

```
> Sol1 := (solve(subs(S = S[1],eq4),z));
```

$$Sol1 := \frac{\pi\,x}{2} - 1 + \frac{\sqrt{\pi^2\,x^2 - 4\,\pi\,x}}{2},\ \frac{\pi\,x}{2} - 1 - \frac{\sqrt{\pi^2\,x^2 - 4\,\pi\,x}}{2},\ \frac{\pi\,x}{2} + 1 + \frac{\sqrt{\pi^2\,x^2 + 4\,\pi\,x}}{2},$$
$$\frac{\pi\,x}{2} + 1 - \frac{\sqrt{\pi^2\,x^2 + 4\,\pi\,x}}{2}$$

```
> k := 1:
evalf(subs(pi=k,x=1,-1*diff(Sol1[1],x)/z[0][1]));
evalf(subs(pi=k,x=1,-1*diff(Sol1[2],x)/z[0][2]));
evalf(subs(pi=k,x=1,-1*diff(Sol1[3],x)/z[0][3]));
evalf(subs(pi=k,x=1,-1*diff(Sol1[4],x)/z[0][4]),30);
```

$$-0.1533219187\,10^{-9} + 0.5773502691\,I$$
$$-0.1533219187\,10^{-9} - 0.5773502691\,I$$
$$-0.4472135955$$
$$0.447213595499957939281834733739$$

La solution 4 est la bonne solution, car elle est la seule qui donne une proportion (réél positif). De plus, on retombe sur un résultat de Nebel,qui arrive à 44 % de bases non appariées dans le cas uniforme (quand π=1).

```
> sol4 := simplify(subs(x=1,-1*diff(Sol1[4],x)/z[0][4]));
```

$$sol4 := \frac{\pi}{\sqrt{\pi\,(\pi+4)}}$$

```
> eqfinale := mu = sol4;
```

$$eqfinale := \mu = \frac{\pi}{\sqrt{\pi\,(\pi+4)}}$$

```
> sol := solve(eqfinale,pi);
```

$$sol := -\frac{4\,\mu^2}{\mu^2-1}$$

```
> w := (mu)-> -4*mu^2/(mu^2-1);
```

$$w := \mu \to -\frac{4\,\mu^2}{\mu^2-1}$$

Donc, pour avoir 75 % de bases non appariées, on doit pondérer les lettres c avec un poids :

```
> evalf(w(75/100));
```

$$5.142857143$$

Et pour 5 % de bases non appariées :

```
> evalf(w(5/100));
```

$$0.01002506266$$