UVSQ université PARIS-SACLAY

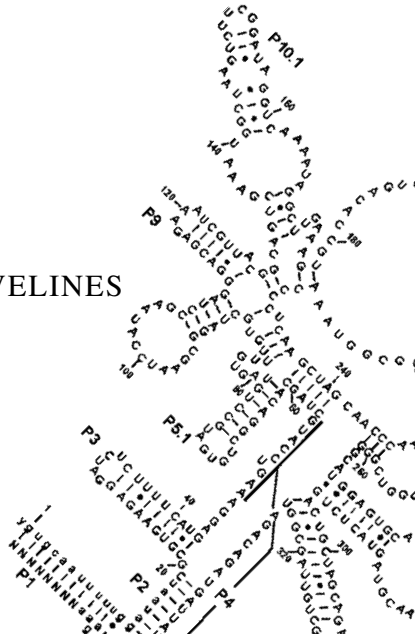
UFR des Sciences
CAMPUS DE VERSAILLES

Rapport de projet sur l'article de recherche et la mise en oeuvre de l'article

Authors
Nathan CARRE, Kubilay MEYDAN

UNIVERSITÉ DE VERSAILLES-SAINT-QUENTIN-EN-YVELINES

May 9th 2023



Contents

1. Introduction

Dans ce rapport, nous allons analyser et développer certaines idées de l'article "A General Edit Distance between RNA Structures" publié en Février 2002 dans le "Journal of Computational Biology: a Journal of Computational Molecular Cell Biology" par TAO JIANG, GUOHUI LIN, BIN MA, and KAIZHONG ZHANG.

L'article aborde plusieurs notions clés de la biologie moléculaire, notamment la notion de structure. En effet, les molécules comme les ARNs peuvent prendre différentes structures : primaire, secondaire et tertiaire. Ces structures peuvent être utilisées afin de caractériser les ARNs et de les comparer entre eux. Ainsi, les auteurs ont proposé différents algorithmes afin de calculer une distance d'édition reflétant la similitude entre deux molécules d'ARNs. Ils ont proposé plusieurs algorithmes pour résoudre différents problèmes d'edit distance, notamment un algorithme de programmation dynamique permettant de calculer la distance d'édition et de faire l'alignement entre deux séquences d'ARNs.

Par ailleurs, nous avons fait le choix d'organiser le rapport de la manière suivante : Partie I : analyse et synthèse des notions abordées dans les parties une, deux et trois de l'article. Partie II : Implémentation de l'algorithme résolvant le problème Edit(nested, plain) abordé dans la troisième partie de l'article. Cette partie propose deux algorithmes permettant respectivement de calculer la distance d'édition et ensuite d'effectuer un alignement entre deux séquences d'ARNs à l'aide d'une méthode de backtracking. Les résultats expérimentaux de notre implémentation sont détaillés à la fin de cette partie. Puis, la Partie III reprend la continuité de l'article et contient la suite de l'analyse et de la synthèse des parties quatre, cinq et six de l'article. Enfin, la Partie IV de notre rapport contient la conclusion générale que nous avons à la suite de l'analyse de l'article et de l'implémentation de l'algorithme de programmation dynamique.

2. Analyse et synthèse des premières parties de l'article

Nous avons fait le choix de faire un résumé de l'article car nous estimons que cela aide à mieux comprendre les notions abordées dans notre projet, directement en lien avec l'article. Pour résumer l'article, les auteurs proposent une méthode pour représenter l'information structurelle des séquences d'ARN en utilisant des séquences annotées d'arcs. Les structures secondaires et tertiaires des ARN peuvent être représentées sous forme d'ensembles d'arcs imbriqués et d'arcs croisés, respectivement. Étant donné que les fonctions des ARN sont en grande partie déterminées par leur conformation moléculaire et donc leurs structures secondaires et tertiaires, la comparaison entre les structures des ARN a récemment suscité beaucoup d'attention. Dans cet article, les auteurs proposent la notion de distance d'édition pour mesurer la similarité entre deux structures secondaires et tertiaires d'ARN, en incorporant diverses opérations d'édition effectuées sur les paires de bases et les arcs . Plusieurs algorithmes sont présentés pour calculer la distance d'édition entre deux séquences d'ARN avec diverses structures d'arcs et selon divers schémas de score, soit de manière exacte, soit de manière approximative, avec des performances prouvées. Des tests expérimentaux préliminaires confirment que la définition de la distance d'édition et le modèle de calcul des auteurs font partie des plus raisonnables étudiés à l'époque de l'article.

Dans l'introduction de l'article, les auteurs traitent de l'importance de l'ARN dans les systèmes biologiques et de l'importance de sa structure secondaire et tertiaire dans la fonction moléculaire. Ils présentent différentes méthodes pour représenter et mesurer la similarité des structures de l'ARN, en utilisant des arbres ou des grammaires contextuelles libres stochastiques (SCFG). Ils proposent ensuite une méthode de mesure de la similarité des séquences d'ARN basée sur la distance d'édition, prenant en compte les structures primaires, secondaires et tertiaires. Les opérations d'édition autorisées sont l'insertion, la suppression et la substitution de bases dans la séquence.

Ensuite, les auteurs décrivent les opérations d'édition à effectuer pour comparer deux séquences annotées d'arcs. Les opérations d'édition comprennent des opérations sur les arcs et les bases, telles que l'arc-match, l'arc-mismatch, l'arc-breaking, l'arc-altering et l'arc-removing. Les opérations de base comprennent la base-match, la base-mismatch, la base-deletion et la base-insertion. Les opérations sont utilisées pour calculer la distance d'édition entre deux séquences annotées d'arcs. Puis ils décrivent les coûts des différentes opérations possibles pour éditer des séquences annotées par des arcs. Le but est de trouver la distance minimale d'édition entre deux séquences annotées d'arcs, en utilisant un schéma de score spécifié par les paramètres wm, wam, wd, wb, wa et wr. Ce schéma de score permet de calculer une distance d'édition en fixant, à l'aide des différents paramètres, le coût de chaque opération possible.

Enfin, les auteurs décrivent 6 possibilités que l'on peut rencontrer lorsque l'on veut effectuer la distance d'édition entre deux structures d'ARN. Les 6 possibilités ordonnées du plus complexe au moins complexe sont les suivantes : Edit(crossing, crossing), Edit(crossing, nested), Edit(crossing, plain), Edit(nested, nested), Edit(nested, plain), et Edit(plain, plain). La possibilité Edit(plain, plain) a déjà été résolue et est bien connue des scientifiques notamment par l'auteur K. ZHANG.

Dans la partie 2 de l'article, les auteurs examinent la complexité de calcul de la distance d'édition entre structures d'ARN. Ils montrent que le problème de calcul de la distance d'édition est NP-difficile, ce qui signifie qu'il est peu probable qu'il existe un algorithme efficace pour le résoudre exactement en un temps polynomial. En effet, les problèmes MAX SNP-hard sont une sous-classe des problèmes NP-difficiles, qui eux-mêmes sont une sous-classe des problèmes NP-complets.

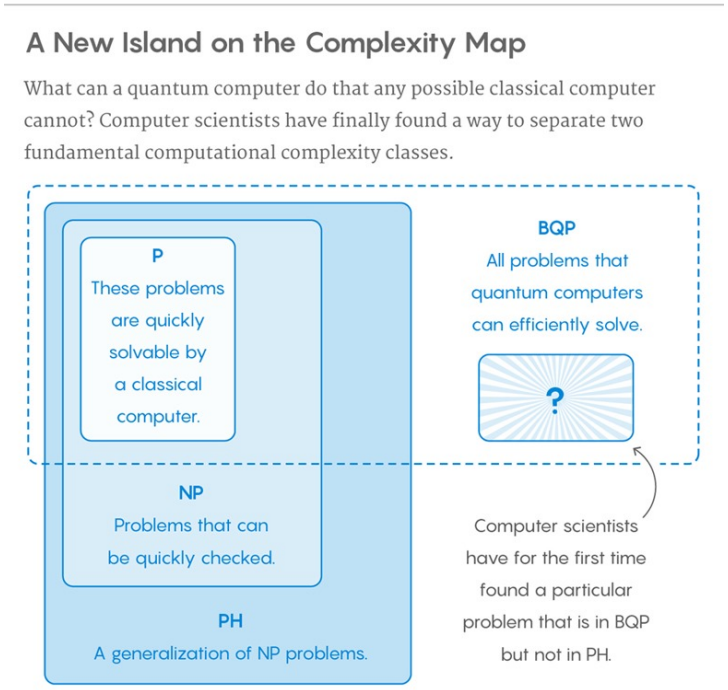


Figure 2.1: Schéma modélisant les principaux ensemble de problèmes, selon leur complexité

Plus précisément, les auteurs montrent que le problème de calcul de la distance d'édition est au moins aussi difficile que le problème de partition, un problème NP-complet bien connu. Il est donc peu probable que le problème de distance d'édition soit résolu en temps polynomial.

Les auteurs montrent également que le problème de décision associé à la distance d'édition est NP-complet. Ils utilisent une réduction à partir du problème du sac à dos, un autre problème bien connu NP-complet, pour montrer que le problème de décision de la distance d'édition est difficile. Cette complexité théorique souligne la nécessité de développer des algorithmes efficaces pour calculer une approximation de la distance d'édition, qui peut être utilisée dans des applications pratiques en biologie computationnelle.

La partie 3 de l'article présente un algorithme de programmation dynamique efficace pour résoudre le problème Edit(nested, plain), qui consiste à comparer une structure secondaire d'ARN à une séquence primaire d'ARN. Les techniques et idées développées dans cette partie peuvent également être utiles pour comparer des structures secondaires et tertiaires d'ARN. Les auteurs présentent une relation de récurrence pour le calcul de la distance d'édition et décrit comment cette relation peut être utilisée dans l'algorithme de programmation dynamique.

L'algorithme calcule la distance d'édition entre les paires de séquences partielles (S1, P1)

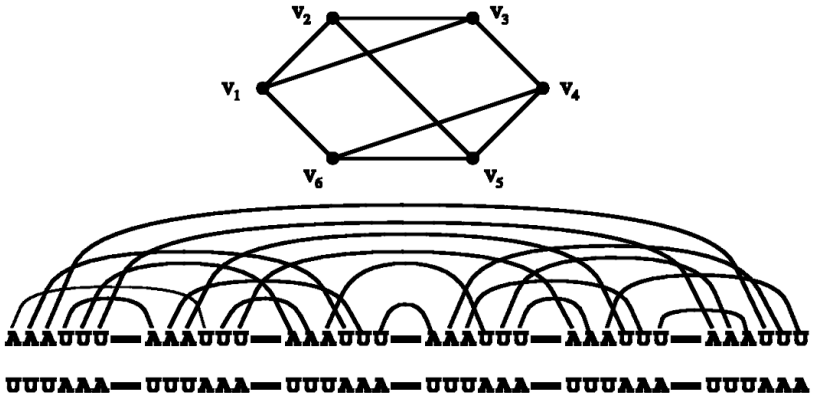


FIG. 2. *L*-reduction: instance construction.

Figure 2.2: Schéma modélisant le problème de partition d'un graph
Jiang et al. (2002)

et (S2, P2) ainsi que l'alignement de séquence optimal. La relation de récurrence de programmation dynamique est décrite pour les cas où les indices sont disjoints et sont définis par des sous-cas, qui incluent les différences de bases et l'ajout, la suppression ou la modification d'arcs dans les structures secondaires. L'algorithme calculant Edit(nested, plain) fonctionne sous n'importe quel schéma de score et a une complexité temporelle de $O(nm^3)$, selon un théorème des auteurs.

Les différents sous-cas étudiés et cités précédemment sont présentés en pseudo-code dans l'article :

Cas 1. Pour tout $1 \leq i \leq n$ et $1 \leq j \leq j' \leq m$,

$$DP(i, i-1; j, j') = (|j' - j| + 1)W_d$$

(où W_d est le poids d'une suppression)

Cas 2. Pour tout $1 \leq i \leq i' \leq n$ et $1 \leq j \leq m$ tel que (i) $i = 1$ ou $(i-1, i'')$ appartient à P_1 pour un certain $i'' > i'$ et (ii) chacun des arcs de P_1 contient entièrement l'intervalle $[i; i']$, ou est contenu en elle, ou est disjoint d'elle,

$$DP(i, i'; j, j-1) = (|P_1[i, i']|)w_r + (i' + 1 - i - 2|P_1[i, i']|)w_d.$$

Cas 3. Pour tout arc (i, i') de P_1 et $1 \leq j \leq j' \leq m$,

$$DP(i, i'; j, j') = \min \left\{ \begin{array}{cc} DP(i+1, i'-1; j+1, j'-1) + w_b + (\chi(i, j) + \chi(i', j'))w_m & \text{si } j < j' \text{ (rupture d'arc)} \\ DP(i+1, i'-1; j, j'-1) + w_a + \chi(i', j')w_m & \text{(modification d'arc)} \\ DP(i+1, i'-1; j+1, j') + w_a + \chi(i, j)w_m & \text{(modification d'arc)} \\ DP(i+1, i'-1; j, j') + w_r & \text{(suppression d'arc)} \\ DP(i, i'; j, j'-1) + w_d & \text{(suppression de base)} \\ DP(i, i'; j+1, j') + w_d & \text{(suppression de base)} \end{array} \right\}$$

Cas 4. Pour tout $1 \leq i \leq i' \leq n$ et $1 \leq j \leq j' \leq m$ tel que (i) (i,i') n'appartient pas à $P1$, (ii) $i = 1$ ou $(i-1,i'')$ appartient à $P1$ pour un certain $i'' > i'$ et (iii) chacun des arcs de $P1$ contient entièrement l'intervalle $[i,i']$, ou est contenu en elle, ou est disjoint d'elle, nous distinguons deux sous-cas:

Si i' est libre, alors

$DP(i,i';j,j') = \min\{DP(i,i'-1;j,j'-1)+\chi(i',j')wm, DP(i,i'-1;j;j')+wd, DP(i,i';j;j'-1)+wd\}$.

Si i' n'est pas libre, c'est-à-dire que $i' = u(i')r$ par la condition (iii) ci-dessus, alors

$DP(i,i';j,j') = \min_{j \leq j'' \leq j'}\{DP(i,u(i')l-1;j,j''-1)+DP(u(i')l,i';j'',j')\}$.

3. Implémentation de l'algorithme (Nested, Plain)

Suite à plusieurs problèmes dans l'implémentation d'autres algorithmes, nous avons finalement décidé d'implémenter l'algorithme (nested plain) qui permet de comparer une séquence primaire et sa structure secondaire a une autre séquence sans sa structure secondaire.

Explications de chaque étape :

1. Initialisation de la table DP : Création d'une matrice DP de dimension (n+2, n+2, m+2, m+2) et initialisation de tous les éléments à l'infini (inf).
2. Définition des valeurs de score : Attribution des valeurs de score aux opérations de base-mismatch (wm), de base-deletion (wd), d'arc-altering (wa), d'arc-breaking (wb) et d'arc-removing (wr). Ces scores ont été définis à la fin de l'article. Nous avons testé plusieurs valeurs, et les scores d'arcs peuvent être mis à zéro pour comparer seulement les séquences primaires.
3. Définition de la fonction de mismatch: Cette fonction retourne 1 si les bases comparées à l'indice i dans S1 et l'indice j dans S2 sont différentes, sinon elle retourne 0.
4. Initialisation de la table DP : Remplissage de la table DP avec les valeurs de base selon les formules de la récurrence (Cas 1) pour les diagonales de la matrice DP.
5. Boucle principale : Pour chaque taille de sous-séquence k (de 1 à n), pour chaque indice i de la séquence primaire, pour chaque indice j de la séquence secondaire et pour chaque indice j' de la séquence secondaire allant de j à m.
6. Cas 3 : Si l'intervalle [i,i'] appartient à la structure secondaire P1, les formules de la récurrence pour le Cas 3 sont utilisées pour calculer DP(i, i'; j, j').
7. Cas 4 : Si i' n'est pas libre et qu'elle est la fin d'un arc dans P1, alors les formules de la récurrence pour le Cas 4 sont utilisées pour calculer DP(i, i'; j, j').
8. Retour de la valeur de DP(1, n, 1, m), qui est la distance d'édition optimale entre les deux séquences S1 et S2.
9. Affichage de la distance d'édition entre S1 et S2.

Nous avons, à la fin de l'algorithme, la distance d'édition entre S1 et S2.

Voici quelques exemples:

Exemple 1:

Séquence 1:
5'- GGAAACAGAAGGACACAGUU -3'
(extrait de human miRNA-23a miRBase accession number: MIMAT0000078)

Séquence 2:
5'- UGAAACAGAAGGAGACAGUG -3'
(extrait de human miRNA-23b miRBase accession number:MIMAT0004513)

Structure P1 de S1:
(.....)..(.....).
Autrement dit,
(1, 8), (13, 20)
obtenu grace a RNAfold web server
(http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi)

Edit distance calculé :
→ 4.0

Exemple 2:

Séquence 3:
5'- UCGGUAUGAGGAUUGCUAUG -3'
(extrait de l'ARNt Phe chez Escherichia coli base de données de l'ARNt chez Escherichia coli de l'Université de Californie, Santa Cruz (UCSC) Genome Browser)

Séquence 4:
5'- CGCUGAGUGACAAAGCAUGC -3'
(extrait de l'ARN non-codant MALAT1 chez l'homme base de données NONCODE)

Structure P1 de S3:
.....().....)
Autrement dit,
(7, 8), (8, 20)
obtenu grace a RNAfold web server
(http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi)

Edit distance calculé :
→ 13.5

Figure 3.1: Exemples 1 et 2 de nos alignements

Pour les deux cas, nous avons utilisé le scoring des auteurs (wm = 1, wd = 1, wa = 1.75, wb = 1.5, wr = 2) et un calcul à la main nous a permis de vérifier le fonctionnement de notre algorithme sur de plus petites séquences, au vu du manque d'exemple des auteurs dans cette articles pour la partie Nested, Plain.

Nous avons ensuite ajouté une fonction backtrace comme l'on fait les auteurs. Cet algorithme n'était pas décrit dans l'article mais est relativement simple (sur une petite matrice). Il consiste à remonter la matrice de distance pour voir les choix de scores pour trouver l'alignement des deux séquences. Nous l'avons implémenté au mieux et nous avons calculé l'alignement de plusieurs séquences. Ces alignements seront ensuite comparés à ceux produits par MUSCLE, l'un des meilleurs algorithmes d'alignement de séquences. (Source: Multiple sequence alignment quality comparison in T-Coffee, MUSCLE and M-Coffee based on different benchmarks Korak et al. (2021)). Ayant déjà incorporé l'algorithme dans le logiciel *A.S.E.N.A.*, nous avons pu obtenir ces résultats:

MUSCLE (3.8) multiple sequence alignment:
S5 U CGGU AU G A G GAUUGCU AUG -
S6 - CGCU GA G U G ACAAAGC AUG C

Notre Algorithme :
S5 U CGGU -A- UGA GG A UU GC U AUG -
S6 - CGCU G A G UGA -C A AA GC - AUG C

MUSCLE (3.8) multiple sequence alignment:
S7 U CG G A U C CAG C UC A G UU GGUAGAG C AGUACGAACC GU AGC G U
S8 A CG U A C C GUU C GA A C UU AAGUCUA C GACGGAUCUA GU CAG G A

Notre algorithme:
S7 U CG G A U CC AGC UC - A GU U GG UA G AG - C - A - G U AC - GA A C -C GU - AG C G U
S8 A CG U A - CC -GU UC G A AC U – UA - AG U C U A C G - AC G GA U C UA GU C AG - G U

En comparant les alignements des algorithmes, nous remarquons que notre algorithme est plus efficace dans l'alignement de séquences similaires (s5, s6). Par contre, sur des séquences avec moins de similarités (s7,s8), notre algorithme tend à trouver des patterns et le meilleur alignement, quitte à ajouter beaucoup de gaps, ce que MUSCLE ne fait pas. Cela peut être changé en augmentant la pénalité des gaps, mais il est important de noter aussi que tous les algorithmes d'alignement n'ont pas tous la même fonction et donc, on ne leur accorde pas les mêmes libertés en termes de manipulation de séquences. Par exemple, certains algorithmes comme MUSCLE vont nous permettre de faire de la phylogénie, en distinguant rapidement les séquences proches et éloignées les unes des autres. On lui attribue donc des pénalités moins fortes en cas de mismatch et plus fortes en cas de gaps. Notre algorithme permettra, lui, d'étudier le lien entre deux ARNs qu'on sait déjà liés, et la transformation éventuelle d'un ARN en un autre. Les mismatchs sont alors coûteux, et les gaps beaucoup moins et l'algorithme a plus de libertés. Il s'agit d'un gradient, et les scores doivent être adaptés à l'utilisation que l'on fait de l'algorithme. Le choix des scores pour les auteurs est justifié par leurs études, pour nous, suivre leurs schémas de score dans notre algorithme était un parti pris subjectif.

L'alignement MUSCLE, le calculateur d'Edit Distance et l'alignement par backtracking sont incorporés dans une suite d'outils bio-informatiques open source que nous sommes en train de développer appelé A.S.E.N.A.

4. Suite de l'analyse et synthèse de l'article

Dans la quatrième partie de l'article, les auteurs décrivent un algorithme pour résoudre le problème de distance d'édition pour les séquences d'ADN avec des arcs imbriqués et croisés. Les auteurs décrivent une classe de schémas de score qui satisfont une condition et qui permettent d'obtenir un algorithme de programmation dynamique efficace. En utilisant cette condition, ils montrent qu'il est possible de simplifier le problème en ne considérant que les opérations de suppression de base, ce qui permet de concevoir un algorithme de programmation dynamique similaire à celui utilisé pour les séquences avec des arcs imbriqués et non-croisés. Les auteurs décrivent ensuite une relation de récurrence pour calculer la distance d'édition en utilisant la programmation dynamique, en distinguant les bases libres des bases incidentes aux arcs. Les auteurs expliquent également comment le coût de la suppression de base est affecté par la présence d'un arc croisé ou imbriqué.

Les auteurs détaillent ensuite ce qu'implique la résolution d'un problème algorithmique lié aux structures d'ARN. Le théorème 4.2 énonce qu'il est possible de trouver une solution approchée poule problème Edit(crossing, nested), avec une complexité de $O(n^3m)$, où n et m sont les tailles des deux structures à comparer. Le corollaire 4.3 précise que le problème Edit(nested, nested) peut être résolu en temps $O(n^3m)$ avec une autre structure de données, qui permet de mieux gérer les performances en pire cas. L'article décrit également un algorithme dans le théorème 4.4 qui améliore la complexité de l'approximation à $O(n^2m)$ du problème Edit(crossing, nested) et permet de calculer une solution en temps raisonnable pour de grandes structures.

Dans la cinquième partie de l'article, les auteurs décrivent les résultats expérimentaux obtenus en utilisant la nouvelle méthode de calcul de la distance d'édition entre deux structures d'ARN. Ainsi, les auteurs ont effectué un test préliminaire de leurs algorithmes pour calculer Edit(nested, nested) sur trois paires d'ARN réels provenant de la base de données RNase P (Brown, 1999). Ils comparent leurs résultats avec ceux de Zhang et al. (1999), qui ont utilisé une hypothèse différente dans leur modèle d'alignement. Zhang et al. (1999) supposent qu'une paire de bases ne peut être appariée qu'avec une autre paire de bases et donc que l'appariement d'une paire de bases avec deux bases non appariées est interdit. Les auteurs proposent un modèle plus réaliste, dans lequel une mutation de bases appariées coûte plus cher qu'une mutation de base non appariée mais moins cher que la suppression d'une paire de bases entière suivie de l'insertion de deux bases non appariées. Les figures 3, 4 et 5 montrent la comparaison des alignements pour les trois paires d'ARN. Dans chaque figure, l'alignement du haut a été produit par l'algorithme de Zhang et al. (1999) et celui du bas par leur propre algorithme sous le schéma de score $(w_r; w_a; w_b; w_a m; w_d; w_m) = (2; 1:75; 1:5; 1:8; 1; 1)$.

Voici ci-dessous les résultats expérimentaux obtenus par les auteurs, suite à trois alignements effectués par leur algorithme d'alignement:

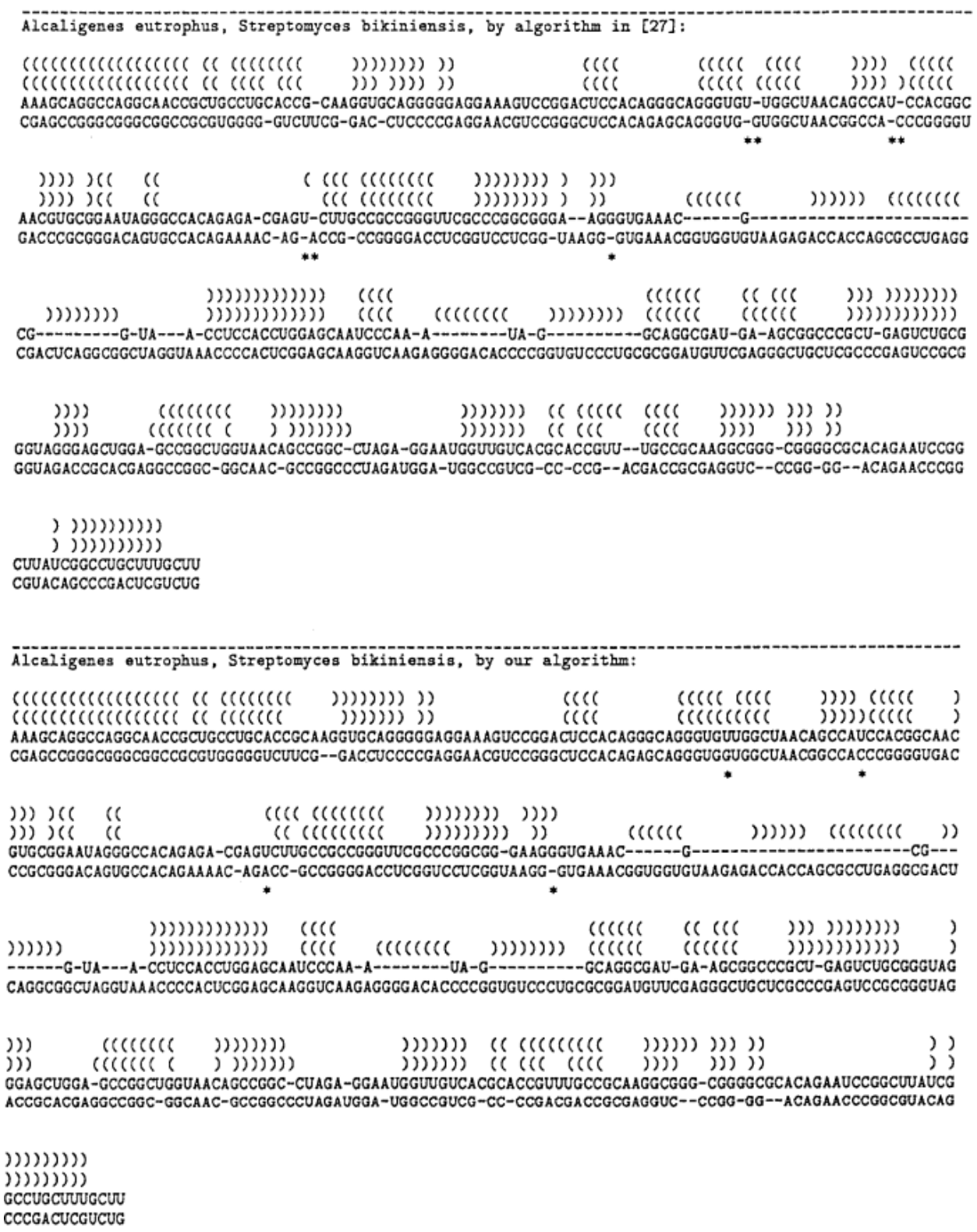


FIG. 3. First Comparison of our alignment and that of a previous algorithm.

Figure 4.1: Première comparaison entre notre alignement et celui d'un algorithme précédent Jiang et al. (2002)

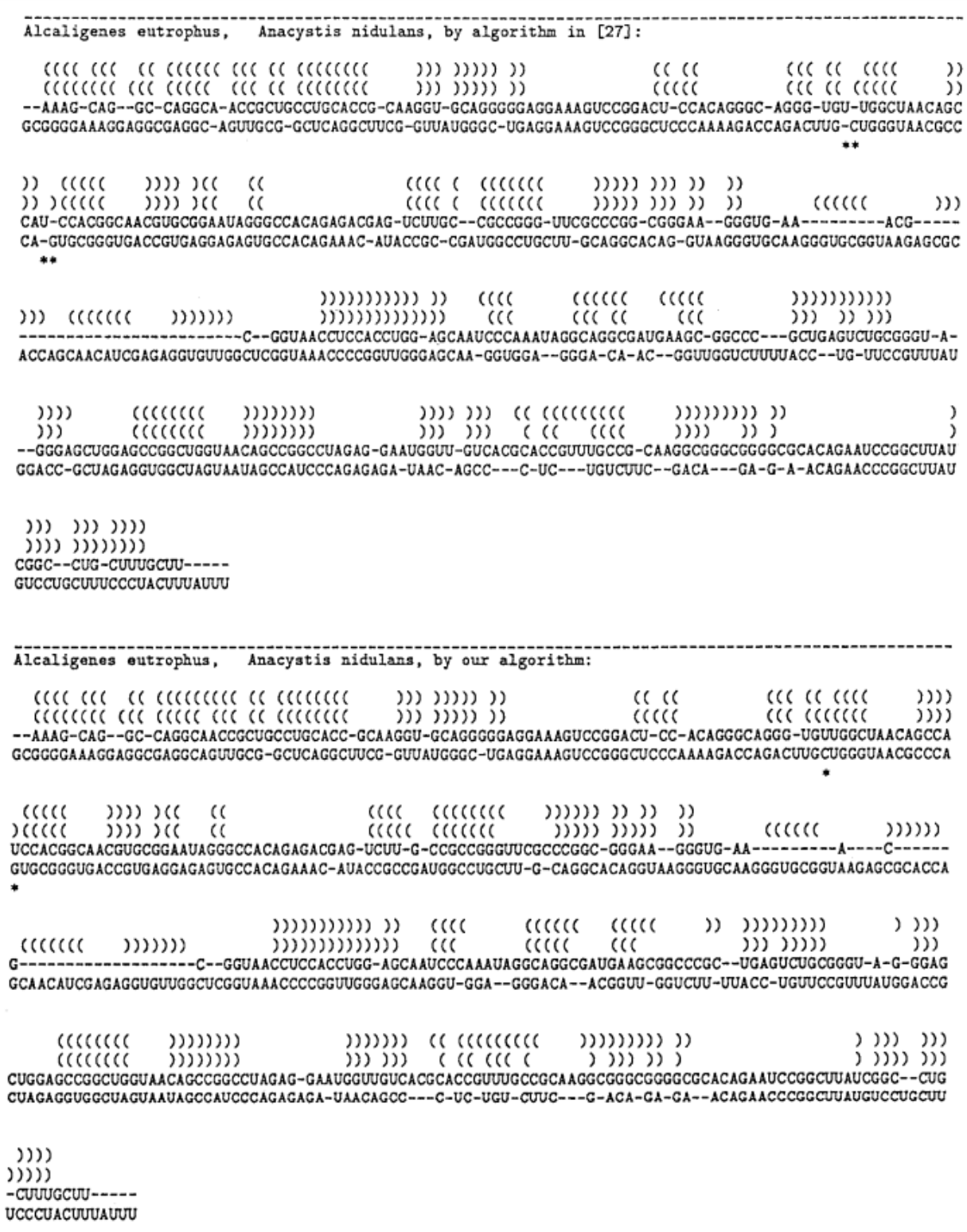


FIG. 4. Second Comparison of alignments.

Figure 4.2: Deuxième comparaison d'alignements
Jiang et al. (2002)

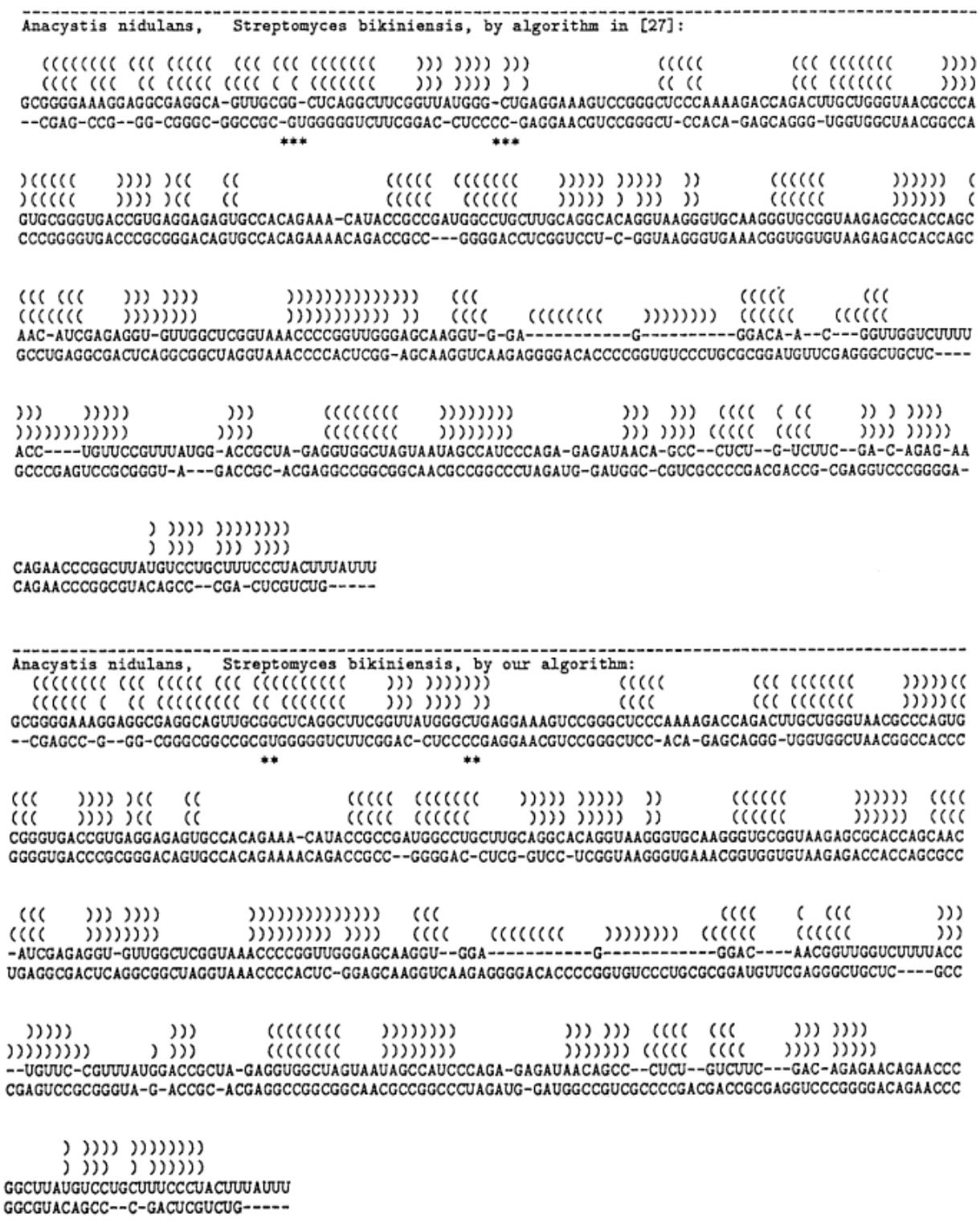


FIG. 5. Third Comparison of alignments.

Figure 4.3: Troisième comparaison d'alignements
Jiang et al. (2002)

Les auteurs ont ainsi trouvé des situations où leur modèle d'alignement est plus réaliste que celui de Zhang et al. (1999).
Dans la dernière partie de l'article, les auteurs concluent l'article et font des suggestions. Ainsi, ils expliquent avoir présenté une nouvelle méthode de calcul d'une distance d'édition pour comparer les structures secondaires et tertiaires de l'ARN. Cette distance prend en compte les opérations sur les paires de bases ainsi que sur les bases individuelles. Les auteurs ont conçu deux algorithmes efficaces pour le calcul de cette distance, l'un pour Edit(nested, plain) et l'autre pour Edit(crossing, nested), sous une classe de schémas de coûts particulier. Ils ont également prouvé que le calcul de cette distance est MAX SNP-difficile pour des schémas de coûts arbitraires.
Les résultats montrent que la distance d'édition définie dans ce document est peut-être plus plausible en pratique que celle d'autres algorithmes existants. Les auteurs suggèrent que l'introduction d'opérations sur les arcs, telles que l'arc-breaking et l'arc-altering, pourrait permettre d'éviter les alignements irréalistes. Ils proposent également de prendre en compte les échanges de bases compensatoires pour mieux refléter les relations structurelles dans la comparaison de structures secondaires de l'ARN.

5. Conclusion

Pour conclure, nous avons vu les méthodes décrites dans cet article pour calculer la distance d'édition entre deux séquences d'arn et leurs structures. Ces méthodes se divisent en deux, les méthodes pour l'étude du problème (nested, plain), et les méthodes pour résoudre les problèmes comparant deux structures: (crossing, nested) et (nested, nested). Ces algorithmes utilisent la programmation dynamique pour avoir un résultat absolu ou approximatif. Nous avons pris la décision d'implémenter l'algorithme nested, plain, en suivant les étapes et le scoring des auteurs. Ce programme a ensuite été incorporé dans notre logiciel de bioinformatique pour offrir une interface intuitive à ces algorithmes, et les mettre en relation avec les autres outils déjà présents dans notre programme. Nos tests sur les sorties de cet algorithme nous ont montré que la pénalité de gaps des auteurs était peut être un peu faible par rapport à la pénalité de substitution. Ce scoring est cependant un choix des auteurs, et convient sans doute à une utilisation précise de l'algorithme. De plus, l'article n'a pas détaillé l'algorithme de backtracking, ce qui peut mener à des confusions dans la vérification de leurs résultats, qu'ils n'ont d'ailleurs pas déclarés pour la partie (nested, plain). Nous avons donc comparé nos résultats à un autre algorithme d'alignement de séquence, MUSCLE. Cet algorithme n'a pas été choisi au hasard, et nous avons cherché des articles comparant les algorithmes d'alignement. L'alignement par MUSCLE est aussi disponible dans le menu phylogénie sur notre logiciel, à la seule exception que l'entrée est en fasta.

L'edit distance est donc une mesure importante, qui sert notamment à aligner les séquences, ce qui est très utile pour faire de la phylogénie, en modifiant les scores. La structure secondaire ayant un impact sur l'édit distance, celle-ci même peut être utilisée pour prédire la structure secondaire des ARN.

Cet article commence cependant à être obsolète, avec des nouvelles techniques de comparaisons d'edit distance qui apparaissent, utilisant un tout autre paradigme de programmation: le deep learning. En 2020, les chercheurs Linfeng Li, Zhipeng Lu, Hao Zhang, Xiaolong Cui et Yifeng Li ont publié un article dans la revue scientifique BMC Bioinformatics décrivant une nouvelle approche d'apprentissage en profondeur pour calculer l'edit distance pour l'ARN. Cette méthode, appelée DeepAlign, utilise un réseau de neurones convolutifs pour apprendre les motifs de séquence locaux et une architecture de réseaux de neurones transformer pour capturer les motifs de séquence globaux et calculer l'edit distance. Sans doute, ces méthodes vont permettre une plus grande compréhension de l'interaction entre la séquence et la structure des ARN.

List of Figures

Bibliography

Jiang, T., Lin, G., Ma, B., and Zhang, K. (2002). A general edit distance between rna structures. *Journal of Computational Biology*, 9(2):371–388.

Korak, T., Aşır, F., Işık, E., and Cengiz, N. (2021). Multiple sequence alignment quality comparison in t-coffee, muscle and m-coffee based on different benchmarks. *Cumhuriyet Science Journal*, 42(3):526–535.