



**PREDICCIÓN DE LA ESTRUCTURA SECUNDARIA DEL ARN
INCORPORANDO MODELOS COMPUTACIONALES DE
EVOLUCIÓN MOLECULAR Y TÉCNICAS DE INTELIGENCIA
ARTIFICIAL**

Manuel Ignacio Cuadrado Morad

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga, Colombia

2016

**PREDICCIÓN DE LA ESTRUCTURA SECUNDARIA DEL ARN
INCORPORANDO MODELOS COMPUTACIONALES DE
EVOLUCIÓN MOLECULAR Y TÉCNICAS DE INTELIGENCIA
ARTIFICIAL**

Manuel Ignacio Cuadrado Morad

Trabajo de investigación presentado para optar al título de:
Magister en Ingeniería de Sistemas e Informática

Directora:

PhD. (c) Lola Xiomara Bautista Rozo

Codirector:

PhD. Francisco José Martínez Pérez

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga, Colombia

2016

Estructurando un sueño...

Agradecimientos

A mi familia cuyo apoyo incondicional me motiva a seguir soñando.

A mi Alma máter, la Universidad Industrial de Santander, por ser un excelente espacio de formación en el cual pude vivir experiencias formidables así como conocer a personas excepcionales.

Al Grupo de Investigación en Ingeniería Biomédica y sus integrantes, por abrirme sus puertas cuando quise ingresar al mundo de la investigación. Una familia en donde me hicieron saber que siempre existirá un lugar para mí.

A mi orientadora, la profesora Lola Bautista, una de las mejores personas que he conocido, fuente de inspiración tanto profesional como personalmente. Su guía y gentileza me permitió culminar con éxito esta aventura.

A mi orientador, el “Sensei” Francisco Martínez, un maestro que siempre busca la excelencia en sus estudiantes. Mil gracias por su confianza y constante motivación.

A mis amigos y compañeros, en especial al “Colegio Cardenalicio” con quienes viví grandes momentos de mi vida.

A los integrantes del Grupo de Procesamiento Paralelo y Distribuido de la Universidad Federal de Rio Grande del Sur, en especial a su director Philippe O. A. Navaux, por recibirme a lo largo de la maravillosa experiencia que fue vivir y estudiar en Brasil.

A todos y cada uno, GRACIAS TOTALES.

Contenido

	Pág.
Introducción	15
Contribuciones del trabajo de investigación.....	16
Organización del documento	16
1. Fundamentación teórica	18
1.1 Ácido Ribonucleico (ARN)	18
1.2 Pares de bases.....	19
1.3 Niveles estructurales del ARN	20
1.3.1 Estructura primaria.....	20
1.3.2 Estructura secundaria.....	21
1.3.2.1 Elementos estructurales.....	23
1.3.3 Estructura terciaria.....	26
1.3.4 Estructura cuaternaria.....	27
2. Predicción de la estructura del ARN.....	28
2.1 Métodos experimentales para la determinación de la estructura del ARN.....	28
2.2 Métodos computacionales para la predicción de la estructura del ARN	29
2.2.1 Predicción de la estructura secundaria por medio de la maximización de pares de bases.....	29
2.2.2 Predicción de la estructura secundaria utilizando la minimización de la energía libre.....	30
2.2.3 Predicción de la estructura secundaria utilizando el modelamiento a partir de Gramáticas.....	30
2.2.4 Predicción de la estructura secundaria utilizando técnicas de inteligencia artificial.....	34
3. Algoritmo de Evolución Gramatical para la predicción de la estructura secundaria del ARN	36
3.1 Algoritmo general	36
3.2 Diseño del algoritmo de Evolución Gramatical	41
3.2.1 Representación	41
3.2.1.1 Representación genotípica.....	42
3.2.1.2 Representación fenotípica.....	43
3.2.2 Gramáticas.	51
3.2.3 Mapeo genotipo a fenotipo.....	52
3.2.4 Función objetivo.....	60

3.2.5 Algoritmo de búsqueda	63
3.2.6 Operadores.....	64
3.2.6.1 Operador de clonación.....	65
3.2.6.2 Operador de hipermutación.....	65
3.2.6.3 Operador de hipermacromutación	65
3.2.6.4 Operador de transmutación.....	66
3.2.6.5 Operador de selección.....	68
3.2.6.6 Operador de envejecimiento.....	68
3.2.6.7 Operador de envolvimiento (<i>wrapping</i>).....	68
3.2.7 Condición de parada.....	68
4. Pruebas y resultados	70
4.1 Primer escenario: Riboswitches	71
4.2 Segundo escenario: ARN de transferencia (ARNt).....	75
4.3 Tercer escenario: Dominios de receptores de Dopamina	79
5. Conclusiones y recomendaciones.....	88
5.1 Conclusiones.....	88
5.2 Recomendaciones.....	89
5.3 Publicaciones.....	89
Bibliografía	91
Anexos.....	99

Lista de figuras

	Pág.
Figura 1. Estructura química de las bases nitrogenadas.	18
Figura 2. Comparación entre los dos tipos de ácidos nucleicos ADN y ARN.....	19
Figura 3. Pares de bases comúnmente observadas en las estructuras del ARN.....	20
Figura 4. Ejemplo de la estructura primaria de una molécula de ARN.	21
Figura 5. Estructura secundaria de la molécula de ARN presentada en la Figura 4 en notación de puntos y paréntesis (Dot-Bracket Notation).....	22
Figura 6. Representación bidimensional de la estructura secundaria.	23
Figura 7. Ejemplo de una estructura secundaria que contiene todos los posibles elementos estructurales.	26
Figura 8. Conformación tridimensional (estructura terciaria) de la molécula de ARN presentada en la Figura 4.....	27
Figura 9. Ejemplos de diferentes árboles de derivación que pueden modelar una misma estructura secundaria.....	33
Figura 10. Conjunto de reglas de producción de una gramática que genera expresiones algebraicas.....	39
Figura 11. Ejemplo de un genotipo en representación binaria.....	42
Figura 12. Representaciones de la estructura secundaria del ARN.	44
Figura 13. Estructura Secundaria de ARN presentada en la Figura 5 en su representación de (a) Puntos y paréntesis, (b) Grammar String, y (c) Grammar Pattern.	45
Figura 14. Estructura Secundaria de la figura anterior y su representación en formas abstractas en los 5 distintos niveles de abstracción.....	46
Figura 15. Ejemplo de una estructura tipo Horquilla conformada por un bucle tipo horquilla y su tallo adyacente.	47
Figura 16. Descomposición de una estructura secundaria en elementos tipo horquilla. .	48
Figura 17. Codificación de la estructura tipo horquilla utilizando los símbolos definidos.	49
Figura 18. Codificación de las horquillas que conforman la estructura secundaria utilizando la notación propuesta.	50
Figura 19. Representación final de la estructura secundaria de la Figura 16 ensamblando los diferentes elementos tipo horquilla.....	50
Figura 20. Representación de una molécula sin plegamiento alguno en notación de horquillas.	51

Figura 21. Trayectoria de plegamiento obtenida al realizar la traducción genotipo-fenotipo.	59
Figura 22. Esquema de la traducción de trayectorias de plegamiento codificadas en cadenas binarias.....	60
Figura 23. Valores de energías para los pares de bases apilados según el modelo del vecino más próximo.	61
Figura 24. Ejemplo de la estimación de la energía libre de una estructura secundaria. ..	63
Figura 25. Ejemplo del operador de hipermutación aplicado a un genotipo binario.	65
Figura 26. Ejemplo del operador de Transmutación aplicado en un genotipo binario de longitud l	67
Figura 27. Ejemplo de la aplicación del operador de envolvimiento.	68
Figura 28. Pseudocódigo del algoritmo opt-aiNET implementado como motor de búsqueda.	69
Figura 29. Resultados generados por Mfold para la secuencia identificada como sbox leader	73
Figura 30. Evolución de la energía libre promedio de toda la población de individuos para el caso de 10 ejecuciones.	74
Figura 31. Estructura secundaria de un ARNt - Phe.	76

Lista de tablas

	Pág.
Tabla 1. Proceso de mapeo para generar una expresión algebraica a partir de un genotipo binario.....	40
Tabla 2. Representación binaria para código gray y código binario.....	43
Tabla 3. Conjunto de operaciones aplicables a estructuras secundarias.....	52
Tabla 4. Ejemplo del proceso de selección de las bases que iniciarán la conformación de un tallo en una estructura secundaria	55
Tabla 5. Estimación de la energía libre de una estructura secundaria escrita en notación de horquillas.....	63
Tabla 6. Conjunto de parámetros establecidos para las pruebas realizadas	71
Tabla 7. Características de las secuencias seleccionadas para el primer escenario de pruebas.....	71
Tabla 8. Resultados obtenidos en el primer escenario de pruebas utilizando Riboswitches.....	72
Tabla 9. Un ejemplo de una de las trayectorias de plegamiento obtenidas al aplicar el algoritmo propuesto a la secuencia identificada como SplicedLeaderAB.....	73
Tabla 10. Resultados obtenidos en el primer escenario de pruebas utilizando un conjunto de secuencias de ARNt.....	76
Tabla 11. Resultados obtenidos para el primer dominio de los receptores de dopamina.....	80
Tabla 12. Resultados obtenidos para el segundo dominio de los receptores de dopamina.....	81
Tabla 13. Resultados obtenidos para el tercer dominio de los receptores de dopamina.....	82
Tabla 14. Resultados obtenidos para el cuarto dominio de los receptores de dopamina.....	82
Tabla 15. Resultados obtenidos para el quinto dominio de los receptores de dopamina.....	83
Tabla 16. Resultados obtenidos para el sexto dominio de los receptores de dopamina.....	84
Tabla 17. Resultados obtenidos para el séptimo dominio de los receptores de dopamina.....	85

Lista de anexos

	Pág.
Anexo A: Gramática para el modelamiento de estructuras secundarias del ARN	99
Anexo B: Ejemplo de la generación de estructuras secundarias de ARN utilizando la gramática propuesta.	102
Anexo C: Gramática para el modelamiento del plegamiento de secuencias de ARN.....	106
Anexo D: Derivaciones del conjunto de operaciones (<i>Move set</i>) definidas para modelar el plegamiento de secuencias de ARN	112
Anexo E: Muestras de las mejores trayectorias de plegamiento encontradas utilizando el algoritmo propuesto.....	119

Resumen

TÍTULO: Predicción de la Estructura Secundaria del ARN Incorporando Modelos Computacionales de Evolución Molecular y Técnicas de Inteligencia Artificial¹.

AUTOR: Manuel Ignacio Cuadrado Morad²

PALABRAS CLAVE: Predicción de la Estructura Secundaria del ARN, Algoritmo Evolutivo, Evolución Gramatical, Minimización de la Energía Libre.

La conformación que las moléculas de ARN adoptan suele estar asociada en gran medida a su función biológica. Debido a esto, la predicción de la estructura secundaria del ARN sigue siendo un problema abierto en el campo de la biología computacional. Aunque existen técnicas experimentales como la Cristalografía de rayos X y la Espectroscopía de Resonancia Magnética Nuclear, que permiten determinar la estructura de este tipo de macromoléculas, su uso suele ser restrictivo debido a su alto costo.

Lo anterior ha impulsado el desarrollo de métodos computacionales que buscan determinar las interacciones que conducen al proceso de plegamiento del ARN.

En este trabajo se presenta un algoritmo evolutivo basado en gramáticas, que permite predecir la estructura secundaria de moléculas de ARN a partir de su secuencia de nucleótidos. El algoritmo propuesto se fundamenta en el concepto de Evolución Gramatical, una técnica de inteligencia artificial originalmente diseñada para generar automáticamente programas en cualquier lenguaje.

Se planteó un modelo de representación que codifica tanto la secuencia de nucleótidos del ARN como su estructura secundaria utilizando elementos estructurales conocidos como horquillas. A su vez se diseñó un conjunto de producciones organizadas en gramáticas que permiten no solo generar estructuras secundarias válidas, sino que también simulan el proceso de plegamiento del ARN a través de la modificación de cadenas definidas bajo el nuevo modelo de representación. Como todo algoritmo evolutivo, se implementa el uso de operadores genéticos con el fin de optimizar una población aleatoria de soluciones potenciales. Cada solución representa una trayectoria de plegamiento, que conduce a una estructura secundaria que la molécula puede adoptar. La energía libre asociada a la estructura obtenida luego de seguir cada posible trayectoria, es utilizada como función objetivo dentro del proceso de optimización. Con cada generación de individuos, las estructuras generadas convergen al estado de mínima energía.

¹ Trabajo de grado

² Facultad de Ingenierías Físico Mecánicas. Escuela e Ingeniería de Sistemas e Informática. Directora: Lola Xiomara Bautista Roza. Codirector: Francisco José Martínez Pérez.

Abstract

TITLE: RNA Secondary Structure Prediction Using Computational Models of Molecular Evolution and Artificial Intelligence Techniques³.

AUTHOR: Manuel Ignacio Cuadrado Morad⁴

KEYWORDS: RNA Secondary Structure Prediction, Evolutionary Algorithm, Grammatical Evolution, Free Energy Minimization.

The conformation that RNA molecules adopt is usually associated to their biological function. Because of this, the prediction of RNA secondary structure remains as an open problem in the field of computational biology. Although there are experimental techniques such as X-ray crystallography and Nuclear magnetic resonance spectroscopy (NMR), for determining the structure of such macromolecules, its use is often restricted due to their high costs.

This has led to the development of computational methods aimed at determine the interactions that lead to RNA folding.

This work introduces a grammar-based evolutionary algorithm that predicts the secondary structure of RNA molecules from its nucleotide sequence. The algorithm is based on the concept of Grammatical Evolution, an artificial intelligence technique originally designed to automatically generate programs in any language.

A model of representation that encodes both RNA nucleotide sequence and its secondary structure using structural elements known as Hairpins was proposed. At the same time, a set of production rules organized in grammars was designed. These rules allow not only to generate valid secondary structures, but also to simulate the folding process of RNA through the modification of strings defined under the new representation model. Like all evolutionary algorithm, the use of genetic operators is implemented in order to optimize a random population of potential solutions. Each solution represents a possible folding path, which leads to a secondary structure that the molecule can adopt. The free energy associated to the structure obtained after following each possible path is used as the objective function in the optimization process. With each generation of evolved individuals, the generated structures converge to the state of minimum energy.

³ Research work

⁴ Faculty of Physical-Mechanical Engineering. Systems Engineering and Informatics Department. Advisor: Lola Xiomara Bautista Rozo. Co-advisor: Francisco José Martínez Pérez.

Introducción

La investigación en Biología Molecular está básicamente orientada a la comprensión de la estructura y funcionamiento de macromoléculas como los ácidos nucleicos y proteínas. Este campo ha sufrido una continua evolución desde el descubrimiento de la estructura del ADN en 1953 (WATSON & CRICK, 1953). De igual forma, con el incremento de la capacidad para manipular secuencias de biomoléculas, se ha generado una vasta cantidad de datos contenidos en repositorios que alojan información genética de numerosas especies.

La necesidad de procesar y analizar este tipo de material, ha redundado en la generación de nuevo conocimiento no solo en el área de la Biología, sino también en disciplinas como las Matemáticas, la Estadística y las Ciencias de la Computación. En este punto se hace importante el término Bioinformática, el cual se puede aplicar de manera general, para referirse al uso y desarrollo de tecnologías de la información aplicadas a la gestión y análisis de datos biológicos (Luscombe, Greenbaum, & Gerstein, 2001), abarcando y articulando desde técnicas relacionadas con la Inteligencia Artificial y la Robótica hasta la Biología Molecular.

La predicción de la estructura de moléculas como el ARN y Proteínas, constituye un problema inmerso en el área de estudio generado por la conjunción de diversas disciplinas como la Genética, La Bioquímica, la Biotecnología, La Biología, la Medicina, la Computación, entre otras, lo cual impulsa de manera constante el desarrollo de investigaciones para abordar esta problemática.

El presente trabajo se enmarca en las áreas de Bioinformática y Computación Natural, siendo esta última un área que abarca el diseño de herramientas computacionales inspiradas en fenómenos de la naturaleza con el fin de dar solución a problemáticas planteadas en diversos entornos. El objetivo de esta investigación se centra en el uso de modelos generativos conocidos como gramáticas para el modelamiento de macromoléculas. A su vez, se explora una solución al problema de la predicción de la estructura secundaria del ARN a partir de su secuencia de nucleótidos aplicando un algoritmo de optimización heurística conocido como Evolución Gramatical, el cual fue originalmente diseñado para la generación automática de programas en cualquier lenguaje definido por una gramática.

Contribuciones del trabajo de investigación

La principal contribución del presente trabajo es un nuevo algoritmo para la predicción de la estructura secundaria de moléculas de ARN a partir de su secuencia de nucleótidos. Este algoritmo integra un conjunto de gramáticas para modelar tanto la estructura secundaria del ARN, como los procesos asociados a la conformación de dichas estructuras; a su vez utiliza un algoritmo evolutivo como metodología de búsqueda para evolucionar, y básicamente encontrar, no solo la estructura óptima para una secuencia, sino también la trayectoria de plegamiento que recorre la molécula antes de alcanzar un estado termodinámicamente estable. Como aportes específicos de este trabajo en relación al modelamiento y predicción de estructuras de ARN, se pueden enunciar los siguientes:

- Se establecieron las características presentes en los fenómenos biológicos asociados al plegamiento de macromoléculas de ácidos nucleicos con el fin de posibilitar la formulación de la predicción de estructuras secundarias como un problema de optimización que puede ser abordado a través de técnicas heurísticas pertenecientes al área de la inteligencia artificial.
- Se identificó un elemento recurrente en toda estructura secundaria que permitió plantear un nuevo sistema de codificación sobre el cual se diseñaron dos modelos generativos, que efectúan una correspondencia entre los procesos biológicos presentes durante el plegamiento del ARN y reglas de producción que operan sobre cadenas de caracteres.
- Se implementó un algoritmo evolutivo, comúnmente utilizado para la programación automática, para realizar el proceso de exploración no solo de la estructura óptima a partir de una secuencia de nucleótidos, sino también la trayectoria de plegamiento que conduce a dicha estructura. El algoritmo presentado integró los modelos gramaticales, y utilizó una función de energía como métrica.
- Se condijeron pruebas para evaluar el desempeño del algoritmo desarrollado, utilizando diferentes conjuntos de secuencias de ARN, obteniendo resultados promisorios.

Organización del documento

Este texto se estructura en los siguientes capítulos:

- El **capítulo 1** describe conceptos básicos del ARN así como los diversos niveles estructurales en los se abstrae este tipo de molécula.

- El **capítulo 2** aborda las razones para la predicción de la estructura secundaria del ARN y describe brevemente algunos métodos de predicción importantes.
- El **capítulo 3** describe la implementación del algoritmo de Evolución Gramatical utilizado para la predicción de la estructura secundaria de moléculas de ARN.
- El **capítulo 4** ilustra las pruebas realizadas a diversas familias de moléculas así como los resultados obtenidos luego de utilizar el algoritmo propuesto.
- El **capítulo 5** presenta las conclusiones del presente trabajo de investigación así como oportunidades para futuros trabajos.
- En los **anexos** se presenta material complementario relacionado con el diseño de las gramáticas utilizadas para modelar la estructura y plegamiento de las moléculas. Detalles asociados a los resultados obtenidos también se despliegan en esta sección.

1. Fundamentación teórica

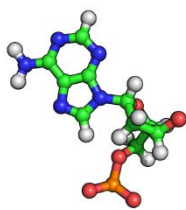
1.1 Ácido Ribonucleico (ARN)

Los ácidos nucleicos son polímeros formados por una cadena repetida de moléculas denominadas nucleótidos. Son los encargados de almacenar la información genética de los seres vivos conocidos. En esencia existen dos tipos de ácidos nucleicos, el ácido desoxirribonucleico, abreviado como ADN, y el ácido ribonucleico conocido como ARN. Este último será el objeto de estudio del presente trabajo de investigación.

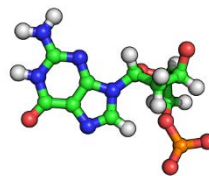
Como se mencionó previamente, los ácidos nucleicos están conformados por una secuencia de nucleótidos que contienen 4 tipos de bases nitrogenadas; en el caso del ADN estos son: Adenina (A), Citosina (C), Guanina (G) y Timina (T). El ARN, similar al ADN, está conformado por los nucleótidos A, C, G y Uracilo (U) que reemplaza a la Timina.

La figura 1 ilustra la estructura química de cada uno de los nucleótidos presentes en los ácidos nucleicos.

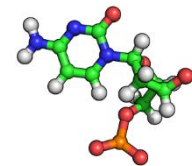
Figura 1. Estructura química de las bases nitrogenadas.



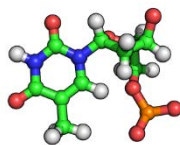
Adenina



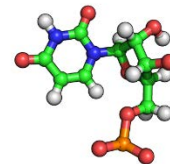
Guanina



Citosina



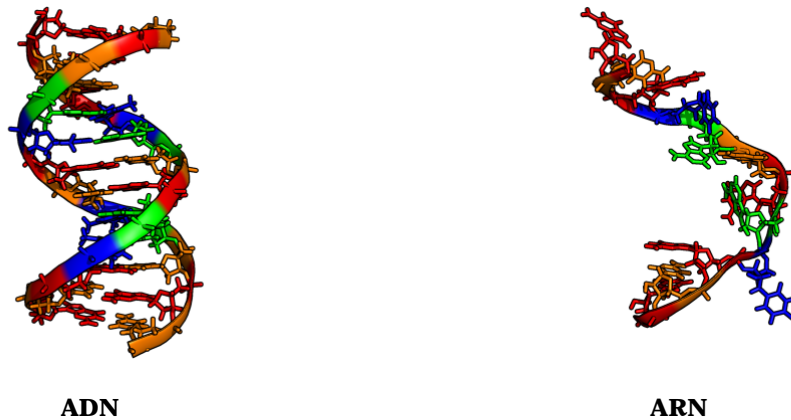
Timina



Uracilo

Una de las principales diferencias entre los dos tipos de ácidos nucleicos radica en que el ADN está compuesto por dos cadenas de nucleótidos, mientras que el ARN está constituido por una cadena sencilla. Por otro lado, uno de los aspectos más importantes de los ácidos nucleicos son los fuertes enlaces formados entre los elementos que los componen; dichas interacciones conocidas como pares de bases permiten que el ADN adopte su característica forma de doble hélice, en la cual una base en una cadena se empareja con una base “complementaria” perteneciente a la cadena opuesta. En el caso del ARN, la molécula tiende a plegarse sobre si misma debido a las interacciones entre sus componentes, formando pares de bases sobre la misma cadena. La figura 2 ilustra la comparación entre las estructuras tridimensionales del ADN y ARN.

Figura 2. Comparación entre los dos tipos de ácidos nucleicos ADN y ARN.



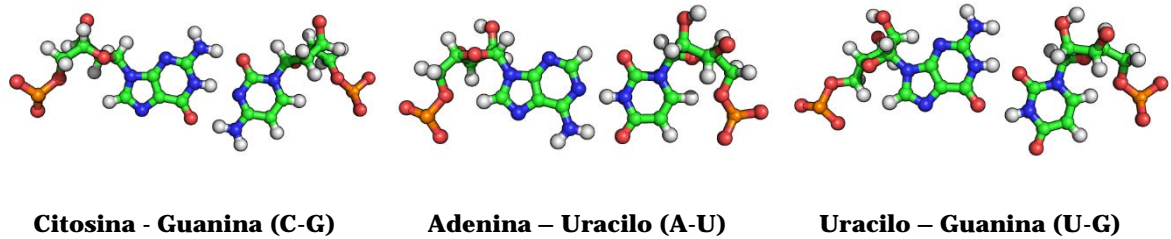
Aunque el ARN desempeña diversas funciones; se le conoce principalmente como el encargado de dirigir las etapas intermedias de la síntesis de proteínas, las cuales son moléculas necesarias para casi todos los procesos biológicos conocidos.

1.2 Pares de bases

Se ha denominado pares de bases a la interacción por puentes de hidrógeno entre una base purínicas (A o G) y una pirimidínica (T, C o U). Los más comunes son los pares de bases Watson-Crick o también denominados apareamientos canónicos, presentes en la doble hélice del ADN y en los emparejamientos entre nucleótidos en moléculas de ARN. Este tipo de pares de bases en el ARN se generan entre bases Adenina-Uracilo y Citosina-Guanina así como una forma no tan estable que se genera entre Uracilo-Guanina conocida como pares

de Wobble. Aunque otras interacciones entre nucleótidos también son posibles, su frecuencia de ocurrencia se considera muy baja. La figura 3 muestra los emparejamientos de bases presentes en las estructuras de moléculas de ARN.

Figura 3. Pares de bases comúnmente observadas en las estructuras del ARN.



1.3 Niveles estructurales del ARN

Como se mencionó previamente, el ARN normalmente existe como una cadena sencilla de bases nitrogenadas. Dicha cadena es lo suficientemente flexible como para que las bases que la componen, interactúen entre sí formando pares de bases; permitiendo a su vez que la molécula se pliegue sobre sí misma adoptando una forma tridimensional, la cual está ligada a la función de la molécula en cuestión.

La estructura del ARN puede ser modelada en cuatro distintas abstracciones jerárquicas que brindan niveles de información asociados a la molécula como se presenta a continuación.

1.3.1 Estructura primaria

La estructura primaria es la manera más simple de representar al ARN; esta describe el orden secuencial de bases que conforman la molécula. Este primer nivel estructural es de utilidad cuando se buscan patrones dentro de la secuencia. Usualmente se escribe como una cadena texto utilizando las letras A, C, G y U, representando cada una de las bases nitrogenadas presentes en el ARN. Por convención, las secuencias de ácidos nucleicos se escriben en la dirección en que son sintetizadas; para esto se adopta el concepto de direccionalidad y se asocia el inicio de la secuencia con el nombre de extremo 5' y el final de la misma con el nombre de extremo 3'. La figura 4 muestra un ejemplo de la representación lineal de la estructura primaria de una molécula de ARN.

Figura 4. Ejemplo de la estructura primaria de una molécula de ARN.

5' -UGCCGAGAGGUCACCUCCGCCACCAUUGGCUCGCAUGUUG- 3'

Formalmente, desde un punto de vista computacional, la estructura primaria del ARN se puede definir de la siguiente forma:

Definición 1:

Sea $\Sigma_b = \{A, C, G, U\}$ el alfabeto que representa el conjunto de bases presentes en el ARN.

La secuencia $S = s_1s_2 \dots s_n$ de n bases s_i donde $1 \leq i \leq n$ y $s_i \in \Sigma_b$, es denominada la estructura primaria o la secuencia de ARN

De ahora en adelante, S denotará una secuencia y n corresponderá a su longitud (número de nucleótidos en la secuencia).

1.3.2 Estructura secundaria

La función de una molécula de ARN está asociada a la forma en la que esta se pliega sobre sí misma; este proceso está determinado por los emparejamientos que acontecen entre cierto número de bases a lo largo de la secuencia. El conjunto de pares de bases que se forman cuando la molécula se encuentra plegada, es lo que se denomina como estructura secundaria.

Definición 2:

Sea S una secuencia de ARN de longitud $n = |S|$.

Una estructura secundaria de ARN de S se define como un conjunto de pares de bases:

$$P = \{(i, j) \mid 1 \leq i \leq j \leq n, S_i \text{ y } S_j \text{ son complementarias}\}$$

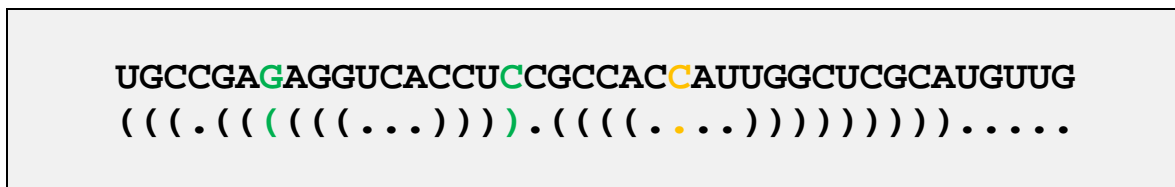
Para todo $(i, j), (i', j') \in P$ donde $i \leq i'$, lo siguiente debe mantenerse:

1. $j - i > 3$
2. $i = i' \Leftrightarrow j = j'$
3. $i < j < i' < j'$ o $i < i' < j' < j$

La regla 1 impone las bases emparejadas deben estar separadas por al menos 3 bases libres; la regla 2 no permite triples interacciones entre pares de bases; y la regla 3 se asegura de que los pares de bases, o se preceden unos a otros, o se encuentran anidados correctamente. Los pares de bases que violan la regla 3, introducen a la estructura elementos denominados pseudonudos (*pseudoknots*); los cuales usualmente no se permiten al momento de modelar la estructura secundaria, a la vez que se clasifican como elementos característicos de la estructura terciaria de la molécula. La tupla (i, j, i', j') , define un pseudonudo si i se empareja con j , i' se empareja con j' , y a su vez se satisface que $i < i' < j < j'$. Los pseudonudos no serán considerados en el presente trabajo.

Cada estructura secundaria puede ser asociada a una secuencia S como una cadena de caracteres B sobre el alfabeto $\Sigma_d = \{ (,), . \}$, donde los puntos representan las bases libres y los paréntesis representan pares de bases; estos últimos deben estar apropiadamente anidados, y tanto B como S deben ser compatibles: si (b_i, b_j) son paréntesis anidados, entonces (S_i, S_j) deben ser un par de bases. Este tipo de representación se conoce como notación de puntos y paréntesis (*Dot-Bracket Notation*). La figura 5 presenta la representación en notación de puntos y paréntesis de la estructura secundaria correspondiente a la secuencia presentada en la Figura 4. Se muestra que la cadena B que representa la estructura secundaria posee la misma longitud que secuencia de nucleótidos S . Adicionalmente se ilustra cómo un par de bases entre un nucleótido en la posición i y otro en la posición j se representa con un paréntesis de apertura en la posición i y un paréntesis de cierre en la posición j de la cadena B ; aquellas bases libres son simbolizadas con puntos.

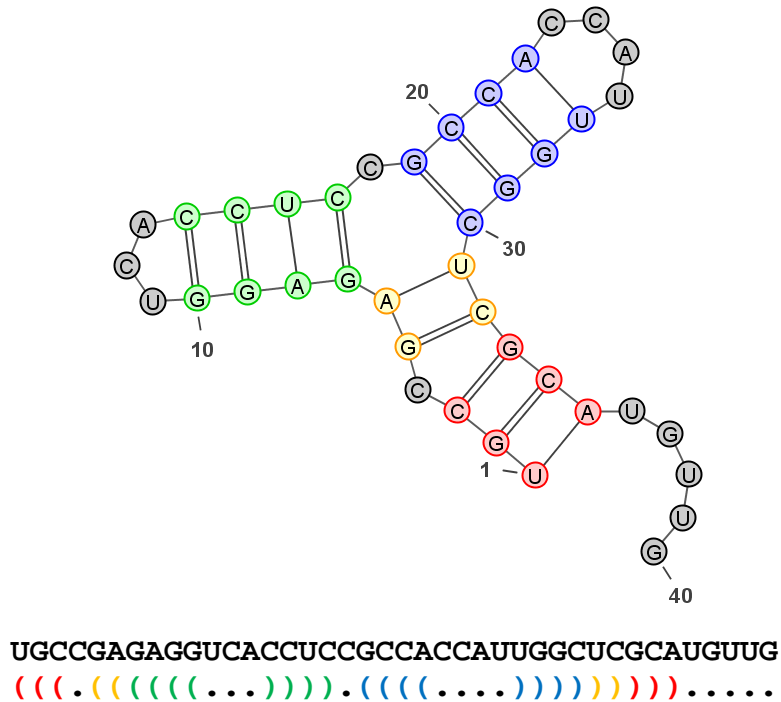
Figura 5. Estructura secundaria de la molécula de ARN presentada en la Figura 4 en notación de puntos y paréntesis (Dot-Bracket Notation)



1.3.2.1 Elementos estructurales

Debido a la naturaleza química de los nucleótidos, cuando existen segmentos complementarios dentro de la secuencia del ARN, los pares de bases tienden a formarse de manera consecutiva, dando lugar a la aparición de dos tipos de regiones dentro de la estructura: Regiones helicoidales dobles denominadas comúnmente como *tallos* y otras regiones conformadas por bases libres denominadas como *bucles* o *burbujas*. La figura 6 muestra la representación bidimensional de la estructura secundaria presentada en la Figura 5 resaltando las regiones correspondientes a los tallos y burbujas.

Figura 6. Representación bidimensional de la estructura secundaria. Se resaltan en colores rojo, amarillo, verde y azul; los pares de bases que conforman los tallos y en color gris, aquellos elementos que conforman los bucles.



Toda estructura secundaria puede descomponerse en varios tipos de elementos recurrentes denominados motivos estructurales. Estos permitirán analizar y describir la estabilidad de la estructura. Los motivos o elementos estructurales considerados en el presente trabajo son los siguientes:

- **Par Apilado (*Stacked pair*):** Están conformados por dos pares de bases consecutivos.

Definición 3:

La dupla (i, j) define un par apilado si i y j se encuentran emparejados y $i + 1$ y $j - 1$ también lo están.

Este tipo de elemento es el único que contribuye a la estabilidad de la estructura secundaria; los demás tipos de elementos estructurales son desestabilizantes en cierta medida. Los *tallos* se generan cuando se presentan dos o más pares apilados consecutivos dentro de la estructura.

- **Bucles tipo horquilla (*Hairpin Loops*):** Contienen exactamente un par de bases adyacente y todas las bases que se encuentren entre las dos bases emparejadas.

Definición 4:

La dupla (i, j) define un bucle tipo horquilla en una estructura secundaria dada, si i y j se encuentran emparejados y k es una base libre $\forall k, i < k < j$.

La estructura conjunta de un bucle tipo horquilla y su hélice adyacente se denomina Horquilla (*Stem-loop*). Las horquillas serán de vital importancia para el desarrollo del presente trabajo y serán nuevamente abordadas posteriormente.

- **Bucles internos (*Internal Loops*):** Es un bucle que contiene exactamente dos pares de bases adyacentes, y todas las demás bases entre ellos se encuentran libres.

Definición 5:

La tupla (i, j, i', j') , en donde $i + 1 < i' < j' < j - 1$, define un bucle interno si i y j se encuentran emparejados, i' y j' se encuentran emparejados y k es una base libre $\forall k, i < k < i' \text{ y } j' < k < j'$.

Los bucles internos se definen según el número de bases libres presentes entre su par de bases adyacentes.

- **Bucle tipo Bulto (*Bulge Loop*):** Es un caso especial de bucle interno en el cual las bases libres se encuentran solo a un lado.
- **Bucle Múltiple o bucle multi-rama (*Multi-branched loops or Multiloops*):** Son bucles unidos al menos a tres pares de bases.

Definición 6:

La tupla $(i, j, i_1, j_1, \dots, i_m, j_m)$, donde $m \geq 2$, $i < i_1 < j_1 < \dots < i_m < j_m < j$ define un bucle múltiple con $m + 1$ ramas; si i se empareja con j , i_1 se empareja con j_1 , i_m se empareja con j_m y k es una base libre $\forall k$, $i < k < i_1, j_1 < k < i_2, \dots, j_m < k < j$.

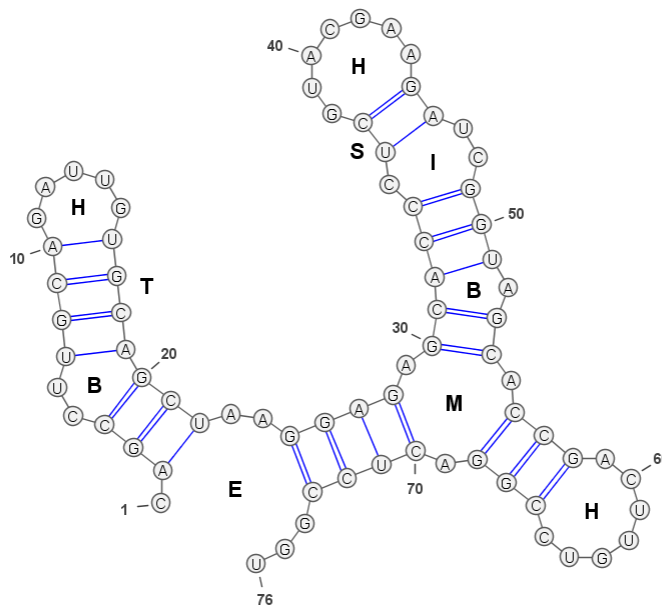
- **Bucle Exterior o externo (*External Loop*):** Son bucles que pueden estar unidos a uno o más pares de bases, y a su vez contienen bases libres en los extremos 5' y 3' de la molécula.

Definición 7:

La tupla $(i, j, i_1, j_1, \dots, i_d, j_d)$, donde $d \geq 0$, $i < j < i_1 < j_1 < \dots < i_d < j_d$ define un bucle externo si i se empareja con j , i_1 se empareja con j_1 , i_d se empareja con j_d y k es una base libre $\forall k$, $1 \leq k < i$, $j < k < i_1, \dots, j_d < k \leq n$, donde n es la longitud de la secuencia.

La figura 7 ilustra un ejemplo de una estructura secundaria en la cual están contenidos todos los posibles elementos estructurales existentes en las moléculas de ARN.

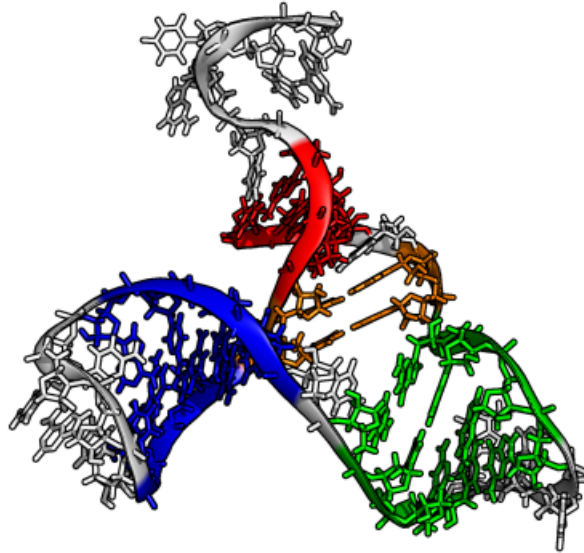
Figura 7. Ejemplo de una estructura secundaria que contiene todos los posibles elementos estructurales. Pares apilados (S), Tallos (T), Bucles tipo Horquilla (H), Internos (I), Bultos (B), Múltiples (M) y Exterior (E).



1.3.3 Estructura terciaria

La estructura terciaria está asociada a las interacciones en el espacio tridimensional de los elementos de la estructura secundaria en una molécula de ARN. Mientras la estructura secundaria brinda información acerca de las bases que interactúan en la molécula, la estructura terciaria describe la ubicación de cada átomo en el espacio. La figura 8 ilustra una representación tridimensional del plegamiento de la molécula presentada en figuras anteriores.

Figura 8. Conformación tridimensional (estructura terciaria) de la molécula de ARN presentada en la Figura 4.



1.3.4 Estructura cuaternaria

La estructura cuaternaria se refiere a las interacciones entre diferentes moléculas de ARN, o entre moléculas ARN y proteínas.

2. Predicción de la estructura del ARN

Uno de los problemas claves de la bioinformática es predecir la estructura tridimensional del ARN a partir de su estructura primaria. Adicionalmente, hoy en día sigue siendo un reto comprender las funciones de una biomolécula como el ARN, las cuales está en gran medida ligadas a la conformación tridimensional de la misma.

Predecir la estructura terciaria de una molécula es un problema computacionalmente complejo. Sin embargo se parte de la premisa que los diferentes niveles estructurales del ARN se conforman de manera independiente pero secuencial. Lo anterior permite tratar la predicción de los mismos de manera separada.

Debido al desarrollo de técnicas de secuenciación que permiten la determinación del orden de los nucleótidos de largas cadenas, la determinación de la estructura primaria de una molécula se considera un problema ya resuelto. Sin embargo, no se puede asegurar lo mismo para los otros niveles estructurales (Krasteva, 2011).

En ese caso, predecir la estructura secundaria resulta importante al momento de brindar información sobre las posibles estructuras que la molécula puede adoptar en el espacio.

El presente trabajo de investigación se centrará precisamente en tratar de resolver el problema de predecir la estructura secundaria de una molécula de ARN a partir de su secuencia de nucleótidos.

2.1 Métodos experimentales para la determinación de la estructura del ARN

Existen diversas técnicas experimentales para la determinación de la estructura de macromoléculas como los ácidos nucleicos y las proteínas. Dentro de las técnicas experimentales más utilizadas se encuentran la Cristalografía de rayos X y la Espectroscopía de Resonancia Magnética Nuclear (NMR). Ambas técnicas proporcionan información específica de la posición, en el espacio tridimensional, de los átomos que conforman a la molécula.

Por lo general, las estructuras obtenidas mediante cristalografía o espectroscopia, han sido recopiladas en un repositorio de dominio público conocido como Protein Data Bank (PDB) (Berman, 2000).

Las técnicas experimentales, sin embargo poseen inconvenientes asociados al costo y al tiempo que consumen (Peignier & Castañeta M., n.d.; Michael Zuker & Sankoff, 1984). Adicionalmente las moléculas se someten a ambientes que muchas veces no son comparables a su estado natural, lo cual afecta el resultado del proceso de predicción.

Debido a eso, se hizo necesario idear métodos computacionales para determinar la estructura secundaria del ARN basados principalmente en su estructura primaria.

2.2 Métodos computacionales para la predicción de la estructura del ARN

Diversidad de métodos computacionales han sido propuestos para resolver el problema de la predicción de la estructura de macromoléculas. No es el objetivo de este capítulo realizar una descripción exhaustiva de estos métodos. Sin embargo se tratará de examinar los principales enfoques aplicados en las técnicas computacionales para la predicción de la estructura secundaria del ARN.

2.2.1 Predicción de la estructura secundaria por medio de la maximización de pares de bases

Generalmente, la predicción de la estructura secundaria de una molécula de ARN se ha abordado como un problema de optimización. Uno de los primeros intentos desembocó en el desarrollo del Algoritmo de Nussinov, el cual busca maximizar la ocurrencia de pares de bases a partir de la secuencia de nucleótidos presente en la molécula (R Nussinov & Jacobson, 1980). La idea sobre la cual se fundamenta este algoritmo es que mientras más pares de bases existan en una estructura, más estable y probable esta será. Para identificar la “mejor” estructura se asignan valores a la ocurrencia o no de un par de bases (Ruth Nussinov, Pieczenik, Griggs, & Kleitman, 1978; Rivas & Eddy, 1999); y se hace uso del método de programación dinámica para resolver el problema de optimización.

Los algoritmos de programación dinámica para la predicción de la estructura secundaria de una secuencia de ARN, funcionan calculando puntajes de plegamientos para todas las subsecuencias $s_i \dots s_j$, partiendo con aquellas de longitud cero, y continúan operando de forma recursiva con secuencias de mayor longitud (Michael Zuker, 2000).

El uso del número de pares de bases como función objetivo para la predicción de la estructura “óptima” resulta un enfoque bastante simplista (Eddy, 2004), que añade limitantes a este método. Entre ellas cabe mencionar que en este enfoque se asume que la ocurrencia de un par de bases es independiente a la conformación o existencia de los otros

pares de bases, desconociendo las interacciones que predisponen a la conformación de regiones de pares de bases adyacentes (tallos). Adicionalmente, aunque el algoritmo de programación dinámica asegura el hallazgo de una estructura matemáticamente óptima, éste no modela de la mejor forma el proceso de plegamiento de una molécula como lo es el ARN produciendo resultados poco realistas.

2.2.2 Predicción de la estructura secundaria utilizando la minimización de la energía libre

Es comúnmente aceptado que en su estado natural las moléculas de ARN se encuentran en conformaciones estructurales con el estado termodinámico más estable, es decir, estructuras que hacen uso de la menor cantidad de energía. En ese sentido, se espera que una molécula de ARN adopte una estructura que minimice la energía para mantenerla estable, y no aquella con el mayor número de pares de bases.

Siguiendo el enfoque que plantea el problema de predicción de la estructura como un problema de optimización; se desarrollaron algoritmos que incluyen información termodinámica en el proceso de predicción.

Dentro de los algoritmos con enfoque “termodinámico”, uno de los más utilizados es el implementado por Zuker (M Zuker & Stiegler, 1981; M Zuker, 1989), el cual, de manera similar al algoritmo de Nussinov, hace uso de la programación dinámica, para explorar el espacio de todas las posibles estructuras secundarias, a la vez que se minimiza una función calculada a partir de parámetros de energía (Akutsu, 2000; Michael Zuker, 2000).

En el caso de los algoritmos de predicción que utilizan una función de energía como función objetivo, se hace necesario contar con parámetros asociados a un modelo termodinámico. Por lo general, estos parámetros son estimados experimentalmente y para luego ser incorporados en el algoritmo al momento de estimar que tan “óptima” es una predicción.

Paquetes de software utilizados para la predicción de la estructura de secuencias de ARN, como *mfold* (M. Zuker, 2003), y *RNAfold* (Hofacker, 2003), implementan el algoritmo de Zuker.

2.2.3 Predicción de la estructura secundaria utilizando el modelamiento a partir de Gramáticas

Según la definición formal de la estructura primaria de una secuencia de ARN; ésta última puede ser considerada como una cadena sobre el alfabeto $\Sigma_b = \{A, C, G, U\}$. En ese caso, resulta natural tratar de modelar al ARN en el contexto de los lenguajes formales (Searls, 1992).

Diversos trabajos se fundamentan en esta idea para implementar modelos basados en gramáticas libres de contexto con el fin de predecir la estructura secundaria de moléculas de

ARN (Nebel & Scheid, 2011; Y Sakakibara et al., 1994), (Wj Anderson, Tataru, Staines, Hein, & Lyngsø, 2012).

Definición 8:

Una gramática libre de contexto G es una tupla (N, T, R, S) , donde N es un conjunto de símbolos no terminales, los cuales no aparecen en las cadenas formadas por G . T es un conjunto de símbolos terminales (un alfabeto), tal que $T \cap N = \emptyset$. R es un conjunto de reglas de producción, de la forma $A \rightarrow \alpha$, $A \in N$, $\alpha \in (T \cup N)^$, donde $*$ es el operador de Kleene. $S \in N$ es el símbolo especial denominado símbolo inicial. Es posible que α sea la cadena vacía; en ese caso se acostumbra a denotar a la cadena vacía con el elemento ε .*

Las gramáticas operan de manera secuencial, en cada paso una regla de producción válida es aplicada sobre un elemento no terminal, transformándolo en una nueva cadena. Este proceso inicia con el símbolo inicial S y prosigue de manera iterativa hasta que se obtenga una cadena conformada solo por elementos terminales.

Definición 9:

Para cualquier cadena $u, v \in (T \cup N)^$ se dice que u produce a v (escrito como $u \Rightarrow v$) si existe una regla de producción $A \rightarrow \alpha$ dentro de R y $u_1, u_2 \in (T \cup N)^*$ tal que $u \rightarrow u_1 A u_2$ y $v \rightarrow u_1 \alpha u_2$. Así v se deriva a partir de la aplicación de la regla de producción $A \rightarrow \alpha$ a u . Si el proceso de derivación de v a partir de u se lleva a cabo en cero o más pasos, se denota $u \xRightarrow{*} v$. Para especificar la gramática utilizada, se denota de la forma $u \xRightarrow[G]{*} v$.*

La derivación de una cadena puede ser ordenada en una estructura tipo árbol denominada árbol de derivación (*parse tree*), el cual brinda una representación de la estructura sintáctica de la secuencia de elementos generados por la gramática. El árbol de derivación juega un papel importante en las gramáticas diseñadas para modelar el ARN debido a que la estructura sintáctica corresponde a la estructura física de la molécula (Yasubumi Sakakibara et al., 1994).

De igual forma se define que un lenguaje generado por una gramática es el conjunto de todas las cadenas que poseen una derivación en esa gramática.

Un ejemplo de una gramática que modela secuencias de ARN sería:

$$N = \{S\}$$

$$T = \{A, C, G, U\}$$

$$P = S \rightarrow aSu \mid uSa \mid cSg \mid gSc \mid \varepsilon.$$

El símbolo \mid se utiliza cuando un elemento no terminal posee más de una regla producción y actúa como un operador OR entre las posibles reglas de producción.

Aplicando de manera aleatoria las reglas de producción de la gramática se puede generar una secuencia de ARN

$$S \Rightarrow aSu \Rightarrow auSau \Rightarrow augScau \Rightarrow auguSacau \Rightarrow augueacau \Rightarrow auguacau$$

Se puede apreciar que la gramática presentada siempre generará secuencias de longitud par cuyos elementos son bases complementarias.

Si se extiende la gramática para que adicionalmente modele bases libres y bifurcaciones, se obtiene el siguiente conjunto de reglas P :

$$S \rightarrow aSu \mid uSa \mid cSg \mid gSc \mid \varepsilon$$

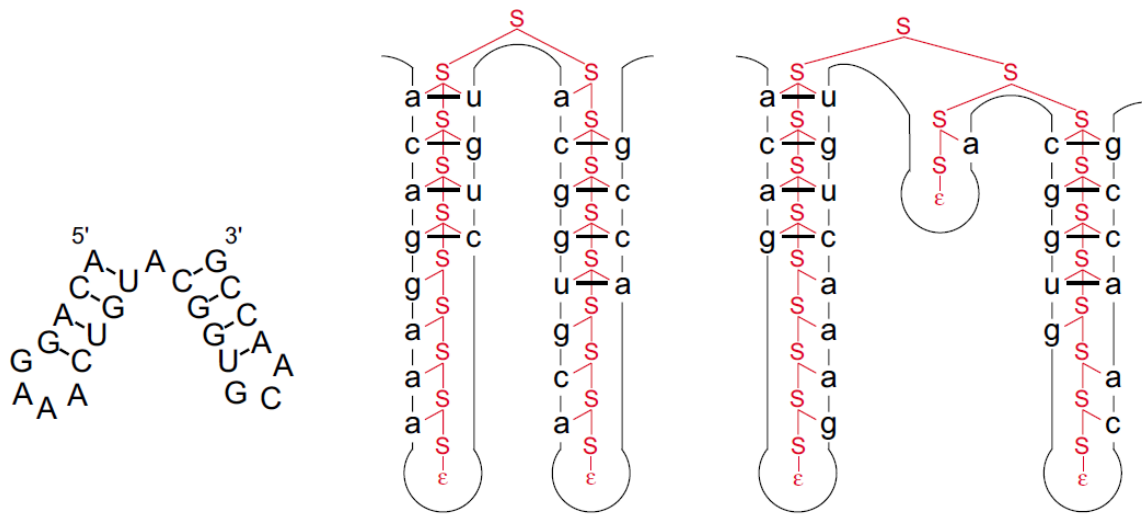
$$S \rightarrow aS \mid uS \mid cS \mid gS$$

$$S \rightarrow a \mid u \mid c \mid g$$

$$S \rightarrow SS$$

Esta gramática puede usarse para generar derivaciones de estructuras secundarias de ARN. Una estructura puede derivarse de múltiples formas dependiendo de las reglas de producción seleccionadas; en todo caso el árbol obtenido será un reflejo de la estructura secundaria de la molécula. La figura 9 ilustra dos casos en los cuales diferentes árboles de derivación, generados a partir de la gramática presentada, pueden modelar la misma estructura secundaria.

Figura 9. Ejemplos de diferentes árboles de derivación que pueden modelar una misma estructura secundaria.



Fuente (Dowell & Eddy, 2004)

En el caso de los algoritmos que hacen uso de una gramática para predecir la estructura secundaria de una secuencia de ARN, no resulta importante conocer si una cadena específica puede ser generada a partir de una gramática dada. Este tipo de técnicas se enfocan más en el proceso de determinar la probabilidad de que una secuencia específica sea generada por una gramática diseñada de tal forma que dentro de sus reglas de producción no solo se modele la secuencia de nucleótidos en una molécula sino también las interacciones que dan lugar a los diferentes elementos presentes en la estructura secundaria (Dowell & Eddy, 2004).

Este enfoque probabilístico al problema de predicción de la estructura es análogo al proceso que realizan los algoritmos en los cuales se maximiza o minimiza una función asociada a la estructura de una molécula, en el sentido que se sigue utilizando una función para evaluar a las estructuras; pero en este caso los parámetros de dicha función son probabilidades, y la estructura “óptima” a determinar, es aquella con la mayor probabilidad de ocurrencia.

El modelo utilizado para llevar a cabo el proceso de predicción es una extensión de las gramáticas libres de contexto denominadas gramáticas libres de contexto probabilísticas (SCFG); las cuales son capaces de asignar una probabilidad a cada cadena que generan (Yasubumi Sakakibara et al., 1994).

Definición 10:

Una gramática libre de contexto probabilística se define como una quintupla (N, T, R, P, S) , donde N es un conjunto de símbolos no terminales. T es un conjunto de símbolos terminales, R es un conjunto de reglas de producción, $P : R \rightarrow (0, 1]$ es una función de probabilidad que asigna una determinada probabilidad a cada regla de producción en R , y S es el símbolo inicial. Para P se debe cumplir para todo elemento no terminal A en N que $\sum_{\alpha}(A \rightarrow \alpha) = 1$. Es decir, la suma de las probabilidades de las reglas de producciones de un mismo elemento no terminal debe sumar 1.

Al asociar probabilidades a las secuencias que genera, una gramática define de manera implícita una distribución de probabilidad sobre el conjunto de secuencias. La probabilidad de un árbol de derivación se calcula como el producto de las probabilidades de las reglas de producción aplicadas para producir la derivación.

La existencia de diversas posibilidades de derivar una secuencia puede generar múltiples árboles de derivación para una misma secuencia obteniéndose a su vez más de una estructura secundaria posible. En ese caso la derivación con mayor probabilidad puede seleccionarse como la estructura secundaria correcta. Sin embargo, debido a que el número de posibles árboles de derivación puede crecer de manera exponencial a medida que la longitud de la secuencia aumenta. Se hace necesario acudir a algoritmos de programación dinámica para evaluar todas las posibles derivaciones. El algoritmo más utilizado en este caso es el denominado algoritmo de Cocke-Younger-Kasami (CYK) (Knudsen & Hein, 2003).

2.2.4 Predicción de la estructura secundaria utilizando técnicas de inteligencia artificial.

Como se mencionó previamente, la predicción de estructuras secundarias puede ser simplificada como un problema de optimización en el cual se minimiza una función de energía; debido a esto esfuerzos recientes se han enfocado en la aplicación de métodos de optimización heurísticos para la exploración del espacio de búsqueda. Técnicas evolutivas como los Algoritmos Genéticos (Deschenes, Wiese, & Poonian, 2004), han sido utilizadas para la predicción de estructuras a partir de la simulación de trayectorias de plegamiento de moléculas de ARN.

Los algoritmos genéticos (AGs) (Goldberg, 1989; J. Shapiro, 2001) son técnicas de optimización estocásticas pertenecientes a la familia de algoritmos evolutivos que operan sobre una población de posibles soluciones. Cada solución se codifica generalmente como un arreglo binario como analogía a la codificación del material genético de todo ser vivo en la naturaleza. Los AGs encuentran una solución modificando de manera aleatoria la población de posibles soluciones utilizando operadores de mutación y recombinando elementos de posibles soluciones “padres” aplicando operadores de cruce, dando así origen

a nuevas soluciones “mejoradas”. A su vez se implementa un criterio de aptitud para seleccionar los mejores individuos y generar una nueva población. La estrategia de evolución de los individuos se repite hasta que no sea posible seguir mejorando la población.

Numerosos trabajos hacen uso de AGs para la predicción de la estructura secundaria del ARN a partir de la secuencia de nucleótidos utilizando la energía libre como criterio de aptitud (Gulyaev, van Batenburg, & Pleij, 1995; B. A. Shapiro, Bengali, Kasprzak, & Wu, 2001), (J. H. Chen, Le, & Maizel, 2000; van Batenburg, Gulyaev, & Pleij, 1995), (B. A. Shapiro, Wu, Bengali, & Potts, 2001). Estos trabajos plantean la predicción de la estructura secundaria como un problema de optimización combinatoria, en el cual se busca “ensamblar” la estructura que contenga la combinación de tallos más estable. El plegamiento de la molécula es simulado como un proceso de transformación entre estructuras intermedias en donde se van agregando de manera gradual tallos a dichas estructuras. Para ello se requiere estimar a priori la lista de todas las regiones complementarias existentes dentro de la secuencia de ARN en las cuales se potencie la formación de pares de bases apilados. A partir de dicha lista se genera una población inicial de estructuras codificadas como arreglos de tallos seleccionados de manera aleatoria. Cada individuo de la población es sometido a operadores de mutación que agregan nuevos elementos de la lista al arreglo (siempre y cuando estos sean compatibles con aquellos agregados previamente) o eliminan elementos del arreglo emulando la disrupción de tallos. También es posible generar nuevos individuos mezclando elementos pertenecientes a dos arreglos distintos, aplicando operadores de cruce.

Otras técnicas de optimización heurística como Optimización por Enjambre de Partículas (PSO) (Geis & Middendorf, 2007; Neethling & Engelbrecht, 2006), (Agrawal & Agrawal, 2015; Geis & Middendorf, 2011) y Optimización por Colonia de Hormigas (Mcmillan, 2006; Yu, Zhang, Liu, & Li, 2010), (Wang, Zhang, Ning, & Chen, 2013), también han sido utilizadas para resolver el problema de predicción de la estructura secundaria del ARN bajo el mismo planteamiento de la optimización combinatoria.

3. Algoritmo de Evolución Gramatical para la predicción de la estructura secundaria del ARN

La Evolución Gramatical (EG) (M. O'Neill & Ryan, 2001; O'Neill & Ryan, 2003) es un tipo de algoritmo bioinspirado, enmarcado dentro del dominio de la Computación Evolutiva, capaz de generar automáticamente programas escritos en cualquier lenguaje de programación definido por una gramática libre de contexto y utilizando una función objetivo que asocia una medida de aptitud a los programas generados. Como todo algoritmo evolutivo, opera sobre un conjunto de posibles soluciones que se van modificando a través de generaciones, permitiendo que aquellas soluciones con mayor potencial, sobrevivan y se reproduzcan. A medida que el algoritmo se ejecuta, la población de soluciones suele converger a la solución real del problema que se está tratando de resolver. En el contexto del ARN, la EG tendrá como objetivo encontrar un conjunto de estructuras secundarias con niveles de energías bajos. Se espera que dentro de las estructuras generadas se encuentre aquella cuyo nivel de energía sea el mínimo.

3.1 Algoritmo general

El modelo biológico sobre el cual se fundamenta la Evolución Gramatical es el proceso de la síntesis de proteínas, el cual inicia a partir del material genético codificado en el genotipo (ADN), que es transcrito a moléculas de ARN, y posteriormente es traducido a una secuencia de aminoácidos, elementos fundamentales en la conformación de proteínas. En el caso de la EG los genotipos son estructuras binarias unidimensionales de longitud variable como analogía a los diversos tamaños de los cromosomas, y los fenotipos son programas codificados en el lenguaje definido por la gramática.

El proceso de generación automática de programas se realiza en dos etapas, la primera constituye la evolución de genotipos y la segunda es la traducción de estos en fenotipos

utilizando las reglas de la gramática que define la sintaxis del lenguaje de programación. En la traducción *genotipo a fenotipo*, el primero codifica la selección de una secuencia de reglas de producción de una gramática libre de contexto. El uso de la gramática asegura que los programas generados sean sintácticamente correctos.

La EG, como cualquier algoritmo evolutivo, opera sobre una población aleatoria de individuos que son mejorados aplicando operadores genéticos, generando así nuevos individuos que a su vez son transformados, haciendo uso de la gramática, en fenotipos a los cuales se les asigna un valor de aptitud asociado a cierta función objetivo. Este proceso se repite generando nuevas poblaciones de individuos hasta que se alcance cierta condición de parada.

Como se mencionó previamente, los genotipos son representados como cadenas binarias de longitud variable haciendo una analogía a las secuencias de nucleótidos en el ADN que posteriormente son “leídos” en tripletas denominadas codones que son los que finalmente codifican los aminoácidos en una proteína. En el contexto de la EG los codones constituyen la transcripción que se realiza al codificar grupos de 8 bits dentro de la cadena binaria en valores enteros que serán utilizados para seleccionar las reglas de producción de una gramática.

La transformación de un genotipo al fenotipo correspondiente se realiza mapeando los codones en reglas de producción de una gramática, y llevando a cabo la proceso de derivación al aplicar las reglas seleccionadas. Como toda derivación, se parte del símbolo inicial definido en la gramática, y se van aplicando las reglas de producción a los elementos no terminales definidos también en la gramática. Existe la posibilidad que sobre un mismo símbolo no terminal se puedan aplicar varias reglas; en este caso dentro del genotipo se utilizan los codones para seleccionar cuál regla se ha de utilizar. En este caso para llevar a cabo la selección de las reglas se utiliza la siguiente expresión:

$$R = C \% NR$$

Donde R es la regla de producción seleccionada, C es un número entero que representa el codón, $\%$ es el operador módulo y NR es el número de reglas que se pueden aplicar al símbolo terminal sobre el cual se operará. Este tipo de mapeo hace que el código utilizado sea degenerado al igual que el código genético en donde cada aminoácido puede ser codificado por más de un codón.

El proceso de transcripción se realiza a lo largo de toda la secuencia de codones. Al finalizar de utilizar el último elemento del genotipo es posible que aún se tengan elementos no

terminales en el programa. En ese caso se puede descartar un individuo representado por el genotipo (asignándole un valor bajo de aptitud), o se puede utilizar un operador característico de la Evolución Gramatical denominado envolvimiento (*wrapping*), que consiste en visualizar el genotipo no como una secuencia lineal de bits, sino como una cadena circular, en la cual se reutilizan los codones leyendo varias veces el genotipo. El operador de envolvimiento permite continuar con el proceso de derivación.

A continuación se presenta un ejemplo de cómo se realiza el proceso de traducción *genotipo a fenotipo* utilizando una gramática de ejemplo

Sea $G = (N, T, P, S)$ una gramática definida para generar cadenas pertenecientes al lenguaje de expresiones algebraicas:

$$N = \{exp, pre_op, op, var\}$$

$$T = \{Sin, Cos, Tan, Log, +, -, /, *, x\}$$

$$S = \langle exp \rangle$$

El conjunto P de reglas de producción se define a continuación en la figura 10.

Figura 10. Conjunto de reglas de producción de una gramática que genera expresiones algebraicas.

$\langle exp \rangle ::=$	$\langle exp \rangle \langle op \rangle \langle expr \rangle$	[0]
	$(\langle exp \rangle \langle op \rangle \langle expr \rangle)$	[1]
	$\langle pre_op \rangle (\langle exp \rangle)$	[2]
	$\langle var \rangle$	[3]
$\langle op \rangle ::=$	$+$	[0]
	$-$	[1]
	$/$	[2]
	$*$	[3]
$\langle pre_op \rangle ::=$	Sin	[0]
	Cos	[1]
	Tan	[2]
	Log	[3]
$\langle var \rangle ::=$	x	[0]

La secuencia de codones generada a partir de la traducción de un genotipo binario se representa por el siguiente arreglo de números enteros

220	202	17	184	107	56	95	218	127	171	189	246	55
-----	-----	----	-----	-----	----	----	-----	-----	-----	-----	-----	----

El proceso de mapeo de la secuencia de codones utilizando la gramática definida se ilustra a continuación en la Tabla 1.

Tabla 1. Proceso de mapeo para generar una expresión algebraica a partir de un genotipo binario.

Fenotipo	Codón	Regla
$\langle exp \rangle$	220	$220 \bmod 4 = 0$
$\langle exp \rangle \langle op \rangle \langle exp \rangle$	202	$202 \bmod 4 = 2$
$\langle pre_op \rangle (\langle exp \rangle) \langle op \rangle \langle exp \rangle$	17	$17 \bmod 4 = 1$
$Cos(\langle exp \rangle) \langle op \rangle \langle exp \rangle$	184	$184 \bmod 4 = 0$
$Cos(\langle exp \rangle \langle op \rangle \langle exp \rangle) \langle op \rangle \langle exp \rangle$	107	$107 \bmod 4 = 3$
$Cos(\langle var \rangle \langle op \rangle \langle exp \rangle) \langle op \rangle \langle exp \rangle$	56	$56 \bmod 1 = 0$
$Cos(x \langle op \rangle \langle exp \rangle) \langle op \rangle \langle exp \rangle$	95	$95 \bmod 4 = 3$
$Cos(x * \langle exp \rangle) \langle op \rangle \langle exp \rangle$	218	$218 \bmod 4 = 2$
$Cos(x * \langle pre_op \rangle (\langle exp \rangle)) \langle op \rangle \langle exp \rangle$	127	$127 \bmod 4 = 3$
$Cos(x * Log(\langle exp \rangle)) \langle op \rangle \langle exp \rangle$	171	$171 \bmod 4 = 3$
$Cos(x * Log(\langle var \rangle)) \langle op \rangle \langle exp \rangle$	189	$189 \bmod 1 = 0$
$Cos(x * Log(x)) \langle op \rangle \langle exp \rangle$	246	$246 \bmod 4 = 2$
$Cos(x * Log(x)) / \langle exp \rangle$	55	$55 \bmod 4 = 3$
$Cos(x * Log(x)) / \langle var \rangle$	220*	$220 \bmod 1 = 0$
$Cos(x * Log(x)) / x$		

* Inicio del proceso de involucramiento

Al finalizar el proceso de derivación se obtiene la expresión válida

$$Cos(x * Log(x)) / x$$

Usualmente, la técnica de Evolución Gramatical hace uso de un Algoritmo Genético como algoritmo de búsqueda (Ryan, Nicolau, & Neill, 2002). Sin embargo al separar el espacio de búsqueda (genotipos) y la solución (fenotipos), es posible utilizar cualquier metaheurística (O'Sullivan, John, 2002), que trabaje sobre representación binaria, como motor de búsqueda. Existen antecedentes del uso de Algoritmos de Optimización por Enjambre de Partículas (O'Neill, Leahy, & Brabazon, 2006; Michael O'Neill & Brabazon, 2006b), (Miranda & Prudêncio, 2015), Evolución diferencial (Michael O'Neill & Brabazon, 2006a) y Algoritmos Inmunes artificiales (Amarteifio & O'Neill, 2005; McKinney & Tian, 2008), (H. S. Bernardino & Barbosa, 2010, 2011b) acoplados a algoritmos de Evolución Gramatical. Los algoritmos de evolución gramatical no solo han sido aplicados a problemas de programación automática (O'Neill & Ryan, 2003), sino que han sido utilizados para abordar problemas de regresión simbólica (H. S. Bernardino & Barbosa, 2011b; Žegklitz & Pošik, 2015) y diseño automático (Ortega, Dalhoum, & Alfonseca, 2003), (N Shaker, 2012; Noor

Shaker, Nicolau, Yannakakis, Togelius, & O'Neill, 2012), (Nasuf, Bhaskar, & Keane, 2013; O'Neill et al., 2010).

3.2 Diseño del algoritmo de Evolución Gramatical

En el presente trabajo se implementa un algoritmo de Evolución Gramatical para resolver el problema de encontrar la estructura secundaria de una molécula de ARN, cuya conformación sea aquella con menor energía, utilizando solo su secuencia de nucleótidos. El proceso de predicción se lleva a cabo a través de la exploración del espacio de todas las posibles estructuras que dicha molécula pueda adoptar. Para ello se hace necesario proponer un sistema de representación o lenguaje que permita simbolizar tanto las estructuras que se generan al plegarse el ARN, como aquellas interacciones que dan lugar a la conformación de dichas estructuras. Estos lenguajes, modelados a partir de gramáticas, permitirán simular la trayectoria de plegamiento (*folding path*) que sigue una molécula desde su estado inicial (*unfolded state*) hasta aquel estado estable de energía mínima.

3.2.1 Representación

La Evolución Gramatical distingue entre dos representaciones asociadas a un mismo individuo, la genética (genotipo) y aquella que es observable (fenotipo). Existe a su vez, un proceso de mapeo que traduce la representación genotípica a la representación fenotípica a través de una gramática. El uso de dos tipos de representación permite que este tipo de algoritmos separe el espacio de búsqueda del espacio de soluciones; la ventaja que se adquiere con esta división es que el motor o algoritmo de búsqueda implementado operará solo en el espacio de búsqueda y podrá incorporar operadores evolutivos genéricos sin importar el contexto del problema. De igual forma, la gramática operará en el espacio de soluciones definiendo un lenguaje que solo es válido en este ambiente, y garantizando que los fenotipos generados a partir de las reglas de producción sean sintácticamente correctos (soluciones válidas). El proceso de mapeo realiza una equivalencia entre genotipos (puntos en el espacio de búsqueda) y fenotipos (elementos en el espacio de soluciones); la medida de qué tan buenos son estos fenotipos (aptitud) se realiza en el contexto del problema.

Para el presente trabajo un genotipo representa una secuencia de operaciones sobre nucleótidos, simulando una trayectoria de plegamiento de una molécula de ARN. De igual forma el fenotipo será la estructura resultante de aplicar las operaciones codificadas en el

genotipo a una secuencia de ARN; en otras palabras la estructura final que adopta la molécula luego de seguir un plegamiento definido.

3.2.1.1 Representación genotípica

Tradicionalmente los algoritmos evolutivos utilizan cadenas binarias de longitud fija para representar individuos en la población de posibles soluciones (Rothlauf, 2006). La Evolución Gramatical utiliza cadenas binarias de longitud variable debido a que estos representan diferentes derivaciones a partir de reglas de producción. Como se mencionó previamente, en el contexto de este trabajo de investigación, los genotipos representan trayectorias de plegamiento codificadas como operaciones sobre nucleótidos de una secuencia de ARN. La trayectoria de plegamiento se define como una serie de operaciones aplicadas sobre nucleótidos que conllevan a la conformación y/o disrupción de estructuras secundarias. En ese sentido la cadena binaria codifica el tipo de operación y la posición de los nucleótidos sobre los que se aplica dicha operación. La figura 11 ilustra un ejemplo de un genotipo en representación binaria.

Figura 11. Ejemplo de un genotipo en representación binaria

01101101	10110011	11110000	...	10101010	11111111
----------	----------	----------	-----	----------	----------

Las operaciones codificadas en el genotipo serán traducidas en reglas de producción traduciendo grupos de 8 bits en números enteros que serán luego utilizados para seleccionar qué reglas aplicar. Existen trabajos previos en el área de algoritmos evolutivos que aseguran que un tipo de representación basada en un sistema de numeración binaria denominada código gray brinda mayores ventajas que la representación binaria tradicional (Hugosson, Hemberg, Brabazon, & O'Neill, 2010). La diferencia entre estos dos tipos de sistemas de numeración es que en el código gray, entre una combinación de dígitos y la siguiente, sólo hay una diferencia de un dígito; lo cual no sucede en el código binario en base 2. La tabla 2 muestra la representación en código binario y código gray de los números del 0 al 9.

Tabla 2. Representación binaria para código gray y código binario

Entero	Binario	Código Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

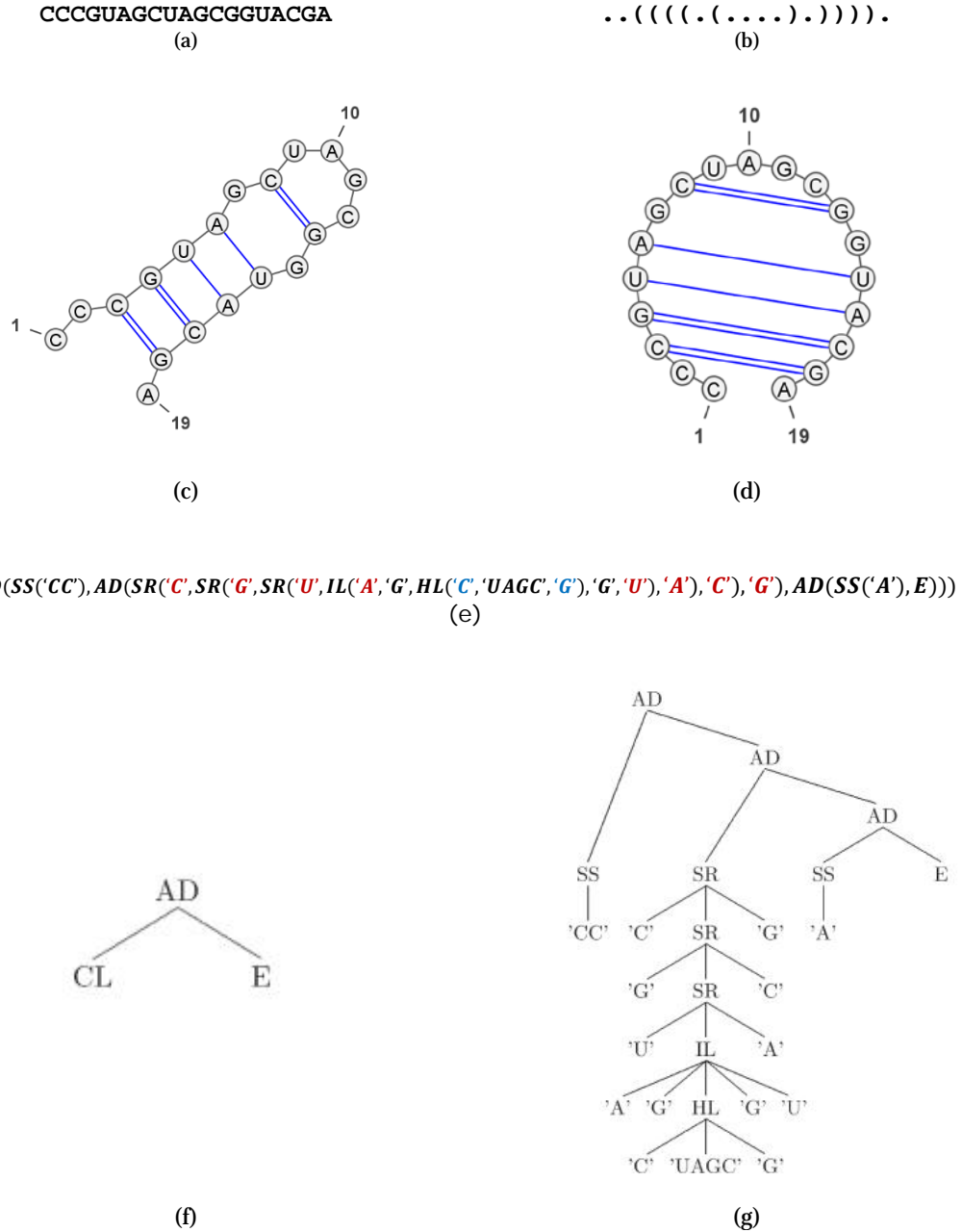
3.2.1.2 Representación fenotípica

El algoritmo de Evolución Gramatical propuesto generará cadenas que representen estructuras secundarias de ARN utilizando una gramática que defina un lenguaje para esta tarea. Usualmente cuando se usan gramáticas para modelar estructuras de ARN, las estructuras se asocian al árbol de derivación.

Existen diferentes formas de representar las estructuras secundarias. Previamente se habían mencionado la representación de puntos y paréntesis y la representación como grafo plano (Fig. 5 y Fig. 6). Existen otros tipos de representación como lo son los grafos circulares y árboles. La figura 12 presenta diversas representaciones de la estructura secundaria del ARN. La desventaja de este tipo de formas de representación es que algunas utilizan elementos distintos para representar secuencia y estructura y otros no facilitan la aplicación de reglas en el sentido estricto debido a que no se trata de cadenas de caracteres sino elementos gráficos. Una gramática se define como un modelo generativo de cadenas de caracteres y uno de los objetivos de este trabajo fue mantener la representación de las estructuras como cadenas de caracteres.

La manera de alcanzar ese objetivo se enfocó en definir un sistema de representación lineal que permitiera codificar información tanto de la secuencia de nucleótidos como de la estructura secundaria de la molécula.

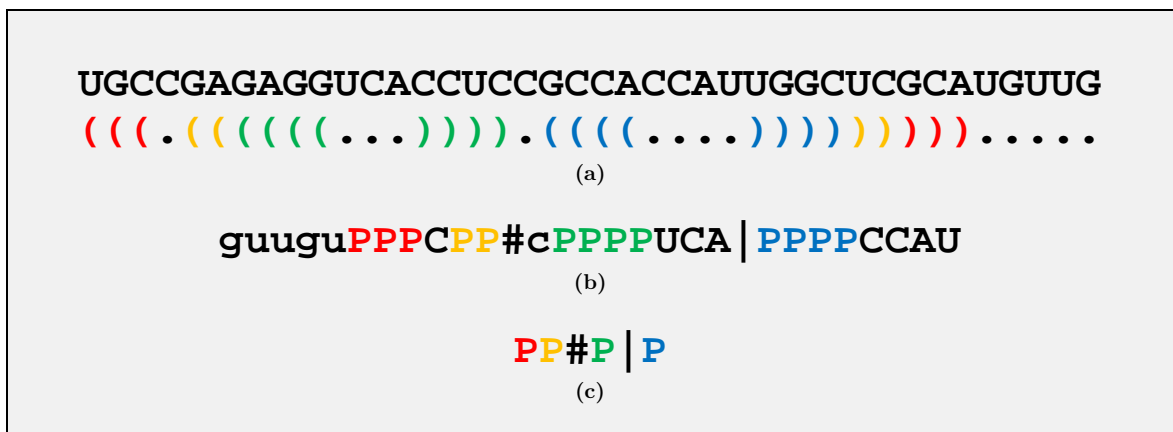
Figura 12. Representaciones de la estructura secundaria del ARN. (a) Estructura primaria. (b) Notación de puntos y paréntesis. (c) Grafo plano (d) Circular. (e) Árbol-ASCII. (f) Forma. (g) Árbol.



Fuente: (Giegerich, 2004)

Trabajos previos abordan el mismo problema al tratar de codificar los dos primeros niveles estructurales en un mismo tipo de representación. Entre estos cabe destacar el tipo de representación conocido como *Grammar Strings* (Achawanantakun, Sun, & Takyar, 2011; Achawanantakun, Takyar, & Sun, 2010) utilizado para codificar secuencias y estructuras secundarias de ARN en cadenas definidas en un alfabeto en donde cada caracter representa una regla de producción de una gramática. Asociado a este tipo de representación también se encuentra otro modelo de codificación conocido como *Grammar Patterns* definido para representar el número de tallos en una estructura y la relación entre estos. Ambos sistemas de representación fueron propuestos para optimizar el alineamiento estructural de secuencias de ARN no codificante. La figura 13 ilustra una estructura secundaria representada utilizando la notación de *Grammar Strings* y *Grammar Patterns*.

Figura 13. Estructura Secundaria de ARN presentada en la Figura 5 en su representación de (a) Puntos y paréntesis, (b) Grammar String, y (c) Grammar Pattern.



Otro trabajo destacable relacionado con la representación de estructuras secundarias es el conocido como Formas Abstractas del ARN (*Abstract Shapes*) (Giegerich, 2004; Steffen, Voss, Rehmsmeier, Reeder, & Giegerich, 2006), (Janssen & Giegerich, 2015), el cual define un sistema de representación en 5 niveles de abstracción ordenados permitiendo comparar estructuras que tengan “formas” similares agrupándolas en clases de estructuras.

A continuación, la figura 14 presenta la estructura secundaria ejemplificada en la figura 13 y su representación utilizando el concepto de formas abstractas del ARN.

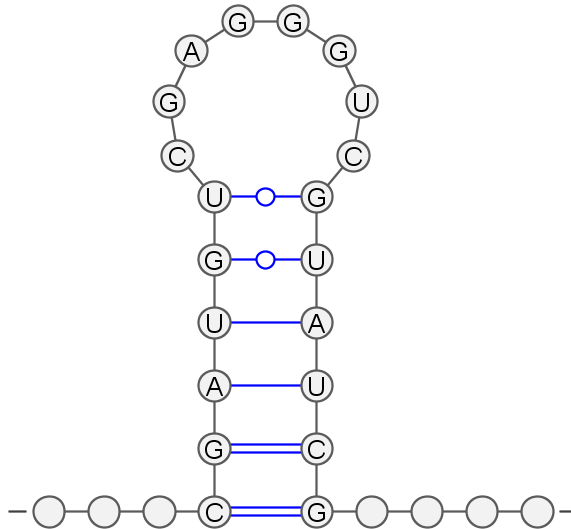
Figura 14. Estructura Secundaria de la figura anterior y su representación en formas abstractas en los 5 distintos niveles de abstracción.



Las desventajas que presenta la representación tipo *String Grammar* es que utiliza un solo símbolo para representar todos los posibles tipos de pares de bases, lo cual dificulta la codificación de las regiones complementarias dentro de la secuencia. A su vez, la posición relativa entre los tallos no es clara. Por otro lado el tipo de representación de Formas abstractas es de utilidad para representar las posiciones relativas de los tallos y bucles en la estructura, pero solo utiliza tres tipos de elementos para codificarlos '[', ']' y '-', haciendo imposible incorporar información relacionada con los nucleótidos de la secuencia.

Para superar estas desventajas se hizo necesario proponer un sistema de representación hecho a la medida del algoritmo implementado. Para esto se partió de la idea de que toda estructura secundaria puede ser descompuesta en elementos estructurales. En este sentido se definió como elemento básico de construcción de toda estructura secundaria, la conformación de un bucle tipo horquilla y su hélice adyacente definida previamente como estructura tipo Horquilla (*Stem-loop*). La figura 15 ilustra un grafo plano representando una estructura tipo horquilla.

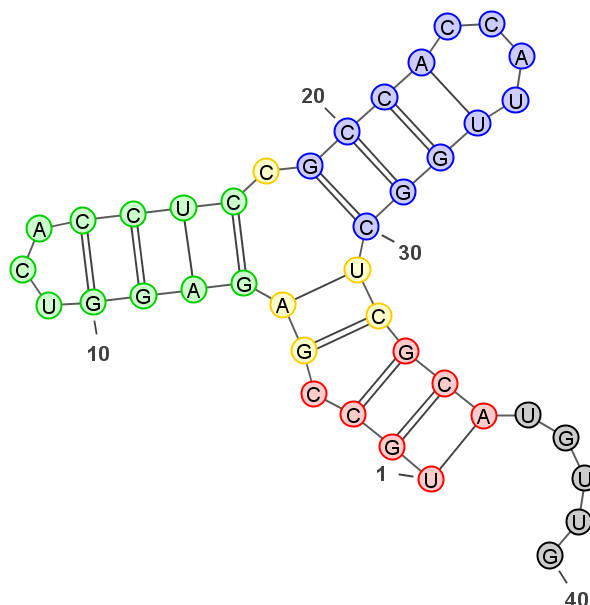
Figura 15. Ejemplo de una estructura tipo Horquilla conformada por un bucle tipo horquilla y su tallo adyacente.



Definido el elemento de construcción elemental, se procedió a definir un alfabeto para representar en cadenas lineales cualquier tipo de estructura horquilla y a su vez representar la relación entre conjuntos de horquillas de tal forma que sea posible ensamblar cualquier estructura secundaria.

De la definición de Formas Abstractas se adoptaron los símbolos '[' y ']' para definir la posición de los extremos 5' y 3' de la horquilla; es decir, los límites de la estructura. De igual forma se hizo necesario definir un elemento que permitiera distinguir dentro de la cadena cuáles elementos hacen parte del tallo y cuales pertenecen a la región del bucle; para esto se definió utilizar el símbolo ':'. La figura 16 presenta la descomposición de una estructura secundaria en elementos tipo horquilla.

Figura 16. Descomposición de una estructura secundaria en elementos tipo horquilla.



Los elementos pertenecientes al tallo de una horquilla deberán codificarse como pares de bases y aquellos pertenecientes al bucle serán codificados como bases libres. Para el caso de los pares de bases se definió un alfabeto de 6 símbolos $\Sigma_p = \{X, Y, Z, W, M, N\}$; con el fin de representar las posibles combinaciones en las cuales los nucleótidos presentes en el RNA pueden emparejarse. De esta forma en cada símbolo se encuentra implícita la información de los nucleótidos que conforman cada par de bases, permitiendo mantener registro de la estructura primaria de la secuencia. Para las bases libres se siguió utilizando el alfabeto de nucleótidos utilizados para definir secuencias de ARN.

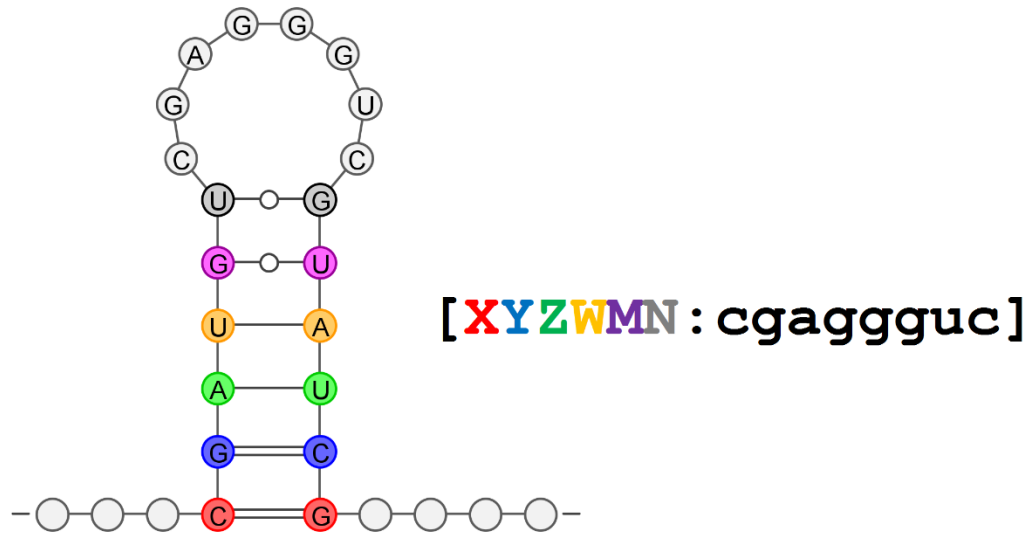
Definición 11:

Sea Σ_{rna} un alfabeto que permite representar cualquier secuencia con regiones complementarias que se plieguen en estructuras tipo horquilla.

$$\Sigma_{rna} = \{c, g, a, u, X, Y, Z, W, M, N, [, :,]\}$$

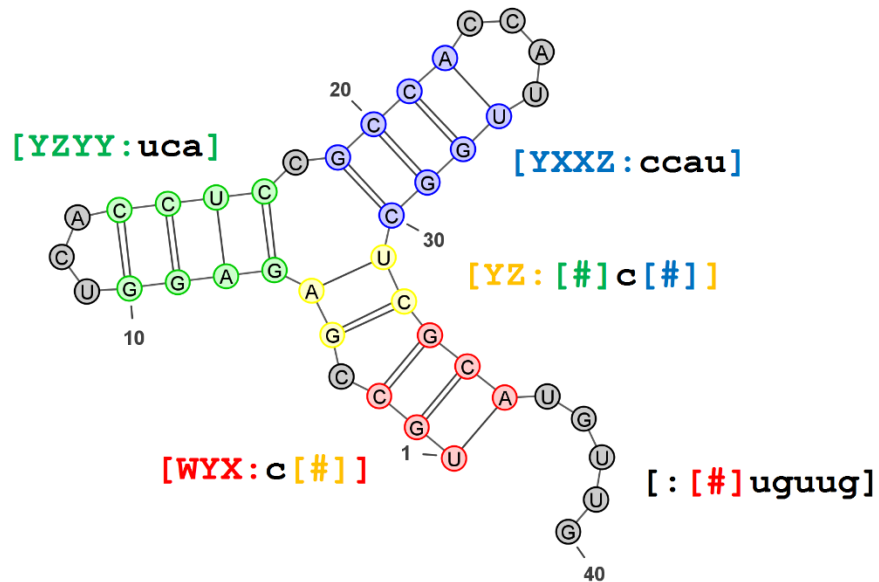
A este tipo de codificación de estructuras secundarias se le denominó notación de horquillas. Un ejemplo de cómo una estructura tipo horquilla es codificada utilizando la notación propuesta es presentada en la figura 17.

Figura 17. Codificación de la estructura tipo horquilla utilizando los símbolos definidos.



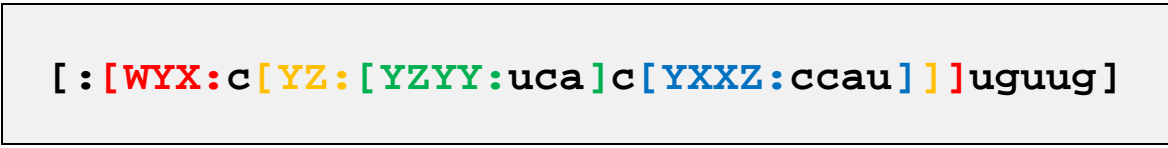
Teniendo un elemento de construcción elemental y un conjunto de reglas para la generación de cadenas usando dicho elemento, es posible ensamblar secuencias que representen estructuras más complejas sobre las cuales se pueda simular el proceso de plegamiento de la molécula aplicando operaciones sobre los símbolos que conforman las cadenas en notación de horquillas. Las figura 18 y 19 ilustran la forma en la que los elementos de una estructura secundaria son codificados y ensamblados utilizando la notación propuesta.

Figura 18. Codificación de las horquillas que conforman la estructura secundaria utilizando la notación propuesta.



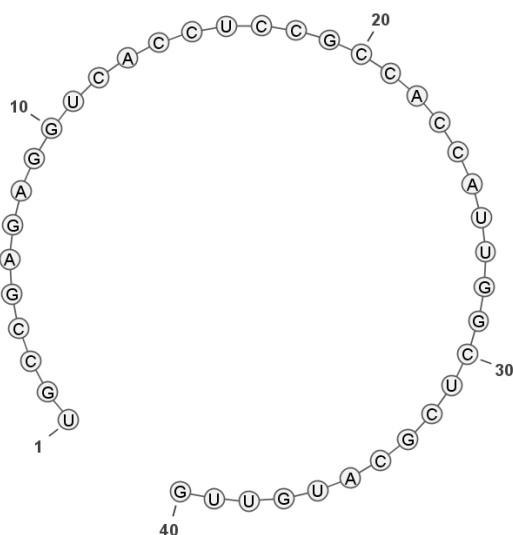
La representación en notación de horquillas permite visualizar los elementos estructurales, su constitución y su posición relativa dentro de la estructura general.

Figura 19. Representación final de la estructura secundaria de la Figura 16 ensamblando los diferentes elementos tipo horquilla.



Una estructura sin pares de bases igualmente puede definirse como una horquilla en la cual el bucle tiene la misma longitud de la secuencia y el tallo está conformado por cero pares de bases. Un ejemplo de lo anterior se muestra en la figura 20.

Figura 20. Representación de una molécula sin plegamiento alguno en notación de horquillas.



[: ugccgagaggucaccuccgccaccauuggcucgcauguug]

3.2.2 Gramáticas

Para el algoritmo de Evolución gramatical implementado se hizo necesario definir dos gramáticas. La primera para modelar la representación de las estructuras secundarias de ARN, y la segunda para organizar un conjunto de operaciones que permitirían simular el proceso de plegamiento de las moléculas.

La primera gramática se planteó para brindar formalismo a la construcción de estructuras secundarias bajo la representación propuesta, denominada previamente como notación de horquillas. Las reglas de producción de esta gramática se construyeron de tal forma que aseguran que las cadenas generadas sean siempre estructuras válidas bajo ese modelo de representación. Dicha gramática se encuentra definida en el Anexo A del presente informe. De manera adicional en el Anexo B se presenta un ejemplo de derivación de una estructura secundaria utilizando la gramática propuesta.

La segunda gramática propuesta, modela operaciones elementales que a su vez conforman un conjunto de movimientos (*move set*), aplicables a estructuras representadas en notación de horquillas. Esto con el fin de simular la formación y destrucción de pares de bases que acontecen a lo largo del trayecto de plegamiento (*folding path*) que recorre la molécula durante el proceso de conformación de las estructuras secundarias finales. Esta gramática

se encuentra definida de manera formal en el Anexo C del presente informe, y un ejemplo de su aplicación se encuentra descrito en el Anexo D.

3.2.3 Mapeo genotipo a fenotipo

La estructura secundaria con menor energía constituye el estado final de una molécula de ARN. El proceso de formación de esta estructura final se define como una secuencia de estados por los cuales la molécula transita antes de llegar a un punto de estabilidad energética. Este conjunto de estados se define dentro del espacio de estructuras que dicha molécula puede adoptar a lo largo del tiempo que dura el plegamiento. La secuencia de estados constituye la trayectoria de plegamiento de la molécula, la cual será definida según el orden en que se presenten las interacciones entre nucleótidos y los lugares específicos en donde estas interacciones inician. Toda trayectoria de plegamiento está conformada por una serie de inserciones y eliminaciones de pares de bases, que inician a partir de un estado sin estructuras, y que dan lugar a la formación de elementos estructurales y interrupción de otros en búsqueda siempre del estado estable final.

En el contexto de la representación de estructuras, las interacciones pueden ser vistas como operaciones sobre símbolos que a su vez representan la dinámica de la formación de las estructuras. Estas operaciones y los elementos sobre los cuales se aplicarán se encuentran codificadas en la representación binaria del genotipo y son mapeadas a estructuras en notación de horquillas a través de las reglas de producción de la segunda gramática propuesta. Para el modelo planteado, se han codificado 5 operaciones o movimientos (*move set*) aplicables a estructuras secundarias con el fin de simular las trayectorias de plegamiento. Este conjunto de operaciones se encuentran enumeradas en la tabla 3.

Tabla 3. Conjunto de operaciones aplicables a estructuras secundarias

Operación	Reglas
Construcción de un tallo	0
Disrupción de un tallo	1
Deslizamiento de un tallo	2
Disrupción de tallos en un mismo nivel	3
Remover un par de bases	4

Un ejemplo de cómo se realiza la transformación de un genotipo en estructuras secundarias se ilustra a continuación:

Sea el siguiente arreglo de números enteros un segmento del conjunto de codones obtenidos al traducir un genotipo binario.

...	25	18	55	129	71	103	...
-----	----	----	----	-----	----	-----	-----

Sea la siguiente secuencia un estado intermedio en el espacio de estructuras que puede adoptar una molécula de ARN

[: [WYX : c [YZ : [YZYY : uca] cgccaccauuggc]] uguug]

Utilizando el genotipo en su representación entera, se procede a leer el primer codón del arreglo para obtener la operación que se aplicará sobre la secuencia.

...	25	18	55	129	71	103	...
	<i>i</i>						

Se aplica la fórmula para seleccionar la operación a aplicar:

$$op = i \text{ mod } (\text{numero de operaciones})$$

$$op = 25 \text{ mod } (5)$$

$$op = 0$$

La operación seleccionada es: **construcción de un tallo**

Un tallo solo puede ser construido dentro de un bucle. Entonces se hace necesario seleccionar en cuál de los bucles presentes en la estructura actual se procederá a construir el tallo. Para esto se procede a utilizar el siguiente codón del genotipo:

...	25	18	55	129	71	...
		<i>i</i>				

El número de bucles presentes en la estructura se obtiene enumerando la aparición del símbolo ':' en la secuencia



Existen un total de 4 bucles dentro de la estructura.

Se procede a aplicar la fórmula para seleccionar el bucle:

$$bucle = i \text{ mod } (\text{numero de bucles}) + 1$$

$$bucle = 18 \text{ mod } (4) + 1$$

$$bucle = 3$$

El tercer bucle ha sido seleccionado



El tercer bucle está conformado por 13 nucleótidos, entonces se dice que la longitud de este bucle es 13.



Se proceden a seleccionar las bases que darán inicio a la conformación del tallo; para esto será necesario leer los siguientes dos codones del genotipo y realizar un proceso análogo a lo que se hizo con la selección del tallo. El proceso de selección de las bases se ilustra a continuación en la tabla 4.

Tabla 4. Ejemplo del proceso de selección de las bases que iniciarán la conformación de un tallo en una estructura secundaria

Selección de la primera base	Selección de la segunda base																												
<table border="1" style="margin: auto;"> <tr> <td>...</td> <td>25</td> <td>18</td> <td>55</td> <td>129</td> <td>71</td> <td>...</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;"><i>i</i></td> <td></td> <td></td> <td></td> </tr> </table>	...	25	18	55	129	71	...				<i>i</i>				<table border="1" style="margin: auto;"> <tr> <td>...</td> <td>25</td> <td>18</td> <td>55</td> <td>129</td> <td>71</td> <td>...</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;"><i>i</i></td> <td></td> <td></td> </tr> </table>	...	25	18	55	129	71	...					<i>i</i>		
...	25	18	55	129	71	...																							
			<i>i</i>																										
...	25	18	55	129	71	...																							
				<i>i</i>																									
$b1 = i \text{ mod } (\text{longitud del tallo}) + 1$	$b2 = i \text{ mod } (\text{longitud del tallo} - 1) + 1$																												
$b1 = 55 \text{ mod } 13 + 1$	$b2 = 129 \text{ mod } (12) + 1$																												
$b1 = 4$	$b2 = 10$																												
<p>[YZ: [#]cgcc<u>acc</u>auuggc]</p>	<p>[YZ: [#]cgcc<u>acc</u>auuggc]</p>																												

Se construye el tallo aplicando la regla de producción que modela esta operación:

$[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgccacc} \text{cau} \text{uggc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgcc} \vec{\text{a}} \text{ccau} \vec{\text{u}} \text{ggc}]] \text{uguug}]$
 $[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgcc} \vec{\text{a}} \text{ccau} \vec{\text{u}} \text{ggc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \vec{\text{c}} [\Psi(\text{a}, \text{u}) : \text{ccau}] \vec{\text{g}} \text{gc}]] \text{uguug}]$
 $[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \vec{\text{c}} [\Psi(\text{a}, \text{u}) : \text{ccau}] \vec{\text{g}} \text{gc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \vec{\text{c}} [\Psi(\text{c}, \text{g}) \text{Z} : \text{ccau}] \vec{\text{g}} \text{c}]] \text{uguug}]$
 $[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \vec{\text{c}} [\Psi(\text{c}, \text{g}) \text{Z} : \text{ccau}] \vec{\text{g}} \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} \vec{\text{g}} [\Psi(\text{c}, \text{g}) \text{XZ} : \text{ccau}] \vec{\text{c}}]] \text{uguug}]$
 $[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} \vec{\text{g}} [\Psi(\text{c}, \text{g}) \text{XZ} : \text{ccau}] \vec{\text{c}}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\Psi(\text{g}, \text{c}) \text{XXZ} : \text{ccau}]]] \text{uguug}]$
 $[: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\Psi(\text{g}, \text{c}) \text{XXZ} : \text{ccau}]]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\text{YXXZ} : \text{ccau}]]] \text{uguug}]$

Se obtiene un nuevo estado intermedio que es representado por una nueva estructura:

[: [WYX:c [YZ: [YZYY:uca] c [YXXZ:ccau]]] uguug]

Se continúa la lectura del genotipo y se selecciona el siguiente codón para escoger la operación a realizar:

...	129	71	103	...
		i		

$$op = i \text{ mod } (\text{numero de operaciones})$$

$$op = 71 \text{ mod } (5)$$

$$op = 1$$

La operación seleccionada es: **disrupción de un tallo**

...	129	71	103	...
			i	

[: [WYX:c [YZ: [YZYY:uca] c [YXXZ:ccau]]] uguug]

* * * * *

$$bucle = i \text{ mod } (\text{numero de bucles}) + 1$$

$$bucle = 103 \text{ mod } (5) + 1$$

$$bucle = 4$$

[: [WYX:c [YZ: [YZYY:uca] c [YXXZ:ccau]]] uguug]

* * * * *

Se procede a suprimir el 4 tallo empezando con el primer par de bases. Este proceso se realiza aplicando la regla de producción definida para esta operación:

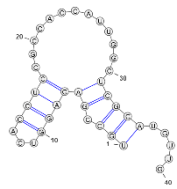
$$\begin{aligned}
& [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\tilde{\text{Y}}\text{ZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
& [: [\text{WYX} : \text{c} [\text{YZ} : [\tilde{\text{Y}}\text{ZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : \phi_5(\text{Y}) [\tilde{\text{Z}}\text{YY} : \text{uca}] \phi_3(\text{Y}) \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
& [: [\text{WYX} : \text{c} [\text{YZ} : \phi_5(\text{Y}) [\tilde{\text{Z}}\text{YY} : \text{uca}] \phi_3(\text{Y}) \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : \text{g}\phi_5(\text{Z}) [\tilde{\text{Y}}\text{Y} : \text{uca}] \phi_3(\text{M}) \text{ccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
& [: [\text{WYX} : \text{c} [\text{YZ} : \text{g}\phi_5(\text{Z}) [\tilde{\text{Y}}\text{Y} : \text{uca}] \phi_3(\text{M}) \text{ccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : \text{ga}\phi_5(\text{Y}) [\tilde{\text{Y}} : \text{uca}] \phi_3(\text{Y}) \text{uccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
& [: [\text{WYX} : \text{c} [\text{YZ} : \text{ga}\phi_5(\text{Y}) [\tilde{\text{Y}} : \text{uca}] \phi_3(\text{Y}) \text{uccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : \text{gag}\phi_5(\text{Y}) \text{uca} \phi_3(\text{Y}) \text{cuccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
& [: [\text{WYX} : \text{c} [\text{YZ} : \text{gag}\phi_5(\text{Y}) \text{uca} \phi_3(\text{Y}) \text{cuccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : \text{gagguca} \text{ccuccg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}]
\end{aligned}$$

Al finalizar la lectura del genotipo se ha alcanzado la estructura final:

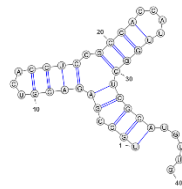
[: [WYX : c [YZ : gaggucaccucc [YXXZ : ccau]]] uguug]

La figura 21 muestra la trayectoria de plegamiento codificada en el genotipo utilizado en el ejemplo.

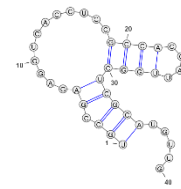
Figura 21. Trayectoria de plegamiento obtenida al realizar la traducción genotipo-fenotipo.



Estructura 0



Estructura 1

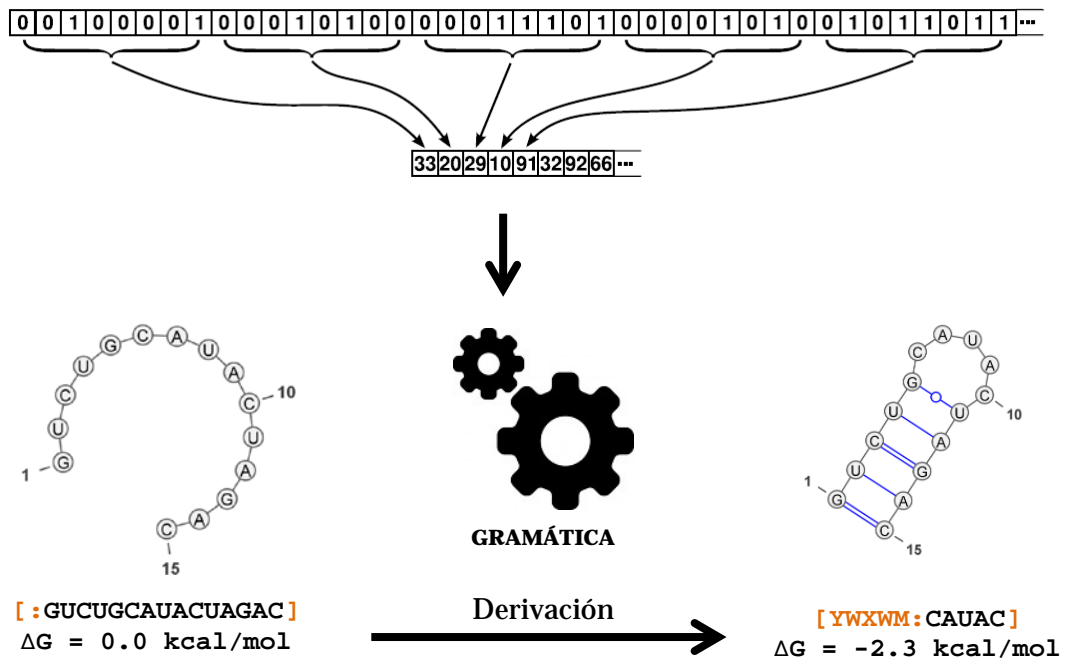


Estructura 2

Cuando un tallo es suprimido de una estructura, la longitud del bucle externo a este se incrementa, lo cual desestabiliza la estructura, y aumenta el valor de energía. Debido a esto, siempre que se realiza una operación que implique la disrupción de tallos, se obliga a que el siguiente movimiento sea la construcción de uno nuevo dentro del bucle formado. Esto garantiza que siempre se busque un nuevo estado con una energía menor al estado anterior. Si no es posible construir un nuevo tallo, o si se construye uno nuevo pero no es lo suficientemente estable para disminuir el valor de energía, el movimiento de disrupción es descartado y se retorna al estado previo. Al descartar codones dentro del genotipo existirán movimientos que no se aplicarán dentro del proceso de traducción del genotipo al fenotipo. Este tipo de fenómeno es análogo a la existencia de segmentos de ADN denominados intrones, los cuales nunca son traducidos al ARN.

Un esquema general de cómo se realiza el mapeo de las trayectorias de plegamiento codificadas en genotipos binarios es ilustrado en la figura 22.

Figura 22. Esquema de la traducción de trayectorias de plegamiento codificadas en cadenas binarias.



3.2.4 Función objetivo

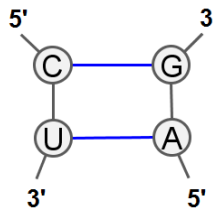
La mayoría de los métodos de predicción de la estructura secundaria del ARN utilizan la energía libre como métrica. Debido a esto, se hace necesario incorporar un modelo termodinámico al momento de calcular las energías libres de las estructuras. Diversidad de modelos termodinámicos han sido incorporados en algoritmos para predecir de manera óptima estructuras secundarias; sin embargo estos modelos requieren de una continua refinación dado que sus parámetros son obtenidos en su mayoría de manera experimental. Algunos modelos son demasiado simples y no logran incorporar todos los parámetros necesarios para calcular las contribuciones a la energía libre. Si los parámetros del modelo no son confiables, la estimación de la energía de las estructuras predichas tampoco lo será. Si bien es conocido que las conformaciones naturales de las moléculas de ARN se presentan en estados bajos de energía, también se acepta que en la naturaleza no siempre la molécula adoptará el estado de energía mínimo (M Zuker, Jaeger, & Turner, 1991). Sin embargo predecir la estructura con el menor valor de energía siempre dará luces de la estructura real de la molécula.

El algoritmo descrito en el presente informe implementa el modelo termodinámico del vecino más próximo *Nearest Neighbour Thermodynamic Model* (NNTM) (Freier et al., 1986), (Douglas H Turner & Mathews, 2010). En este modelo, valores de energía son asignados tanto a grupos de pares de bases apilados como a los bucles formados por bases libres. Bajo este esquema, se asume que la estabilidad de una estructura se concentra en las interacciones entre pares de bases adyacentes al interior de los tallos de la estructura. Por el contrario, los bucles presentes en la estructura contribuyen de manera negativa a la estabilidad de la misma; dichas contribuciones dependerán del tamaño y tipo de bucle. El nivel de estabilidad de la estructura se establece a través del cambio estándar de energía libre ΔG .

El modelo del vecino más próximo plantea que la energía libre de una estructura secundaria específica se calcula sumando las contribuciones de cada motivo estructural presente en la estructura (Mathews, Sabina, Zuker, & Turner, 1999). En ese caso se tienen parámetros de energía para cada combinación de pares de bases y tipos de bucles, los cuales han sido mejorados a lo largo de los años y publicados en un repositorio para uso público (D H Turner et al., 1987), (Jaeger, Turner, & Zuker, 1989), (Xia et al., 1998), (Douglas H Turner & Mathews, 2010). El conjunto de valores de energía se encuentra definido a manera de reglas que se aplican al calcular la energía de la estructura. El conjunto de parámetros de energía utilizados para los pares de bases apilados según el modelo del vecino más próximo se presenta en la figura 23 junto con un ejemplo de la forma en que son aplicados.

Figura 23. Valores de energías para los pares de bases apilados según el modelo del vecino más próximo.

	CG	GC	GU	UG	AU	UA
CG	-2.4	-3.3	-2.1	-1.4	-2.1	-2.1
GC	-3.3	-3.4	-2.5	-1.5	-2.2	-2.4
GU	-2.1	-2.5	1.3	-0.5	-1.4	-1.3
UG	-1.4	-1.5	-0.5	0.3	-0.6	-1.0
AU	-2.1	-2.2	-1.4	-0.6	-1.1	-0.9
UA	-2.1	-2.4	-1.3	-1.0	-0.9	-1.3



-2.4 kcal/mol

El presente trabajo implementó rutinas para estimar la energía libre de una estructura en notación de horquillas a partir de los parámetros del modelo del vecino más cercano. Se seleccionó este modelo debido a que también es implementado por algoritmos ampliamente utilizados como Mfold (Michael Zuker, 2003) Pfold (Knudsen & Hein, 2003) y Vienna (Hofacker, 2003). La figura 24 ilustra un ejemplo de la estimación de la energía libre de una

estructura secundaria a partir de las contribuciones energéticas de cada elemento que la conforma. En la tabla 5 se muestra el mismo proceso de cálculo de energía libre pero en este caso analizando sintácticamente la estructura secundaria en notación de horquillas.

Figura 24. Ejemplo de la estimación de la energía libre de una estructura secundaria. Se muestran las contribuciones energéticas de los pares de bases apilados, así como de los bucles que conforman la estructura.

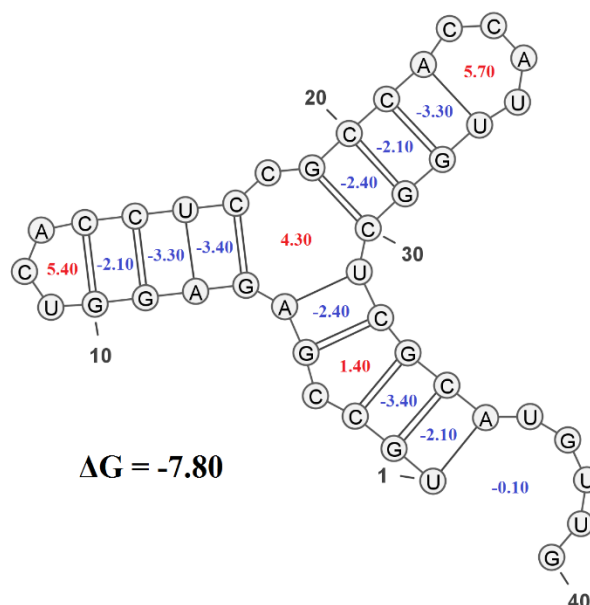


Tabla 5. Estimación de la energía libre de una estructura secundaria escrita en notación de horquillas

Elemento estructural	Tipo	ΔG [kcal/mol]
[: [#] uguug]	Bucle Externo	-0.10
[WYX :	Tallo	-5.50
: c [#]]	Bucle Interior (1,0)	1.40
[YZ :	Tallo	-2.40
: [#] c [#]]	Bucle Múltiple	4.30
[YZYY :	Tallo	-7.80
: uca]	Bucle Horquilla	5.70
[YXXZ :	Tallo	-8.80
: ccau]	Bucle Horquilla	5.40
ΔG Total		-7.80

3.2.5 Algoritmo de búsqueda

Como se mencionó previamente, una de las ventajas de los algoritmos de Evolución Gramatical es que al separar los espacios de búsqueda y solución, es posible utilizar

cualquier metaheurística como motor de búsqueda siempre y cuando esta trabaje sobre una representación binaria.

Dada que la naturaleza del espacio de búsqueda presenta superficies de energías rugosas con múltiples óptimos locales y puntos de inflexión (S.-J. Chen & Dill, 2000; Li & Zhang, 2012), se hace necesario seleccionar como motor de búsqueda, una metaheurística que pueda lidiar con problemas de optimización que presenten este tipo de funciones objetivo.

Dentro de las metaheurísticas utilizadas para optimización, que ya han sido validadas en contextos de optimización multimodal, se encuentran los Sistemas Inmunes Artificiales (AIS) (Leandro N de Castro & Timmis, 2002). Este tipo de algoritmos son modelos computacionales que toman como inspiración la manera en que el sistema inmunológico de los organismos responde a los ataques. Estos algoritmos bioinspirados han sido ampliamente utilizados en el área de optimización (H. Bernardino & Barbosa, 2009), e incluso han sido desarrollados sistemas inmunes basados en gramáticas para resolver diversos problemas de programación automática y regresión simbólica (Amarteifio & O'Neill, 2005; H. S. Bernardino & Barbosa, 2009), (H. S. Bernardino & Barbosa, 2010, 2011a).

Dentro de la diversidad de modelos existentes dentro de los sistemas inmunes artificiales se encuentra una familia de algoritmos denominados redes inmunes artificiales, diseñados en parte para resolver problemas de optimización multimodal (L.N. de Castro & Timmis, n.d.) (de França, Von Zuben, & de Castro, 2005).

Los sistemas inmunes artificiales utilizados en el área de la optimización, al igual que los algoritmos genéticos, simulan el proceso de evolución de una población de individuos que representan posibles soluciones a un determinado problema. Se inicia con un conjunto de individuos generados de manera aleatoria y a partir de estos, se producen nuevas generaciones de individuos por medio de la aplicación de operadores

En el presente trabajo se implementó como algoritmo de búsqueda, una variante de un algoritmo de la familia de redes inmunes artificiales conocido como opt-aiNET (de Franca, Coelho, & Von Zuben, 2010). Con este algoritmo se buscó evolucionar una población de trayectorias de plegamiento de moléculas de ARN codificadas como cadenas binarias.

3.2.6 Operadores

Como todo algoritmo evolutivo basado en poblaciones, las redes inmunes artificiales incorporan una serie de operadores genéticos los cuales son aplicados sobre cada individuo

de la población con el fin de mejorar su aptitud. Los algoritmos basados en la familia de redes inmunes conocida como aiNET aplican mecanismos que ajustan el tamaño de la población y mantienen diversidad de soluciones permitiéndoles explorar de manera más amplia el espacio de búsqueda asociado al problema que se pretende solucionar (de Franca et al., 2010). En el algoritmo propuesto se implementaron operadores de reproducción, mutación, envejecimiento y selección de los individuos.

3.2.6.1 Operador de clonación

A diferencia de los algoritmos evolutivos comunes, en los cuales la generación de nuevos individuos se lleva a cabo a través de operadores de cruce, los sistemas inmunes artificiales modelan un proceso de reproducción asexual como el presente en las células. En ese orden de ideas cada individuo puede reproducirse generando un conjunto de copias idénticas de sí mismo. El número de clones que cada individuo puede criar será proporcional a su aptitud. La medida de aptitud siempre se relaciona con el valor que toma la función objetivo para dicho individuo. En el presente caso se define como el valor de energía libre asociada a la estructura a la cual se llega luego de recorrer la trayectoria de plegamiento.

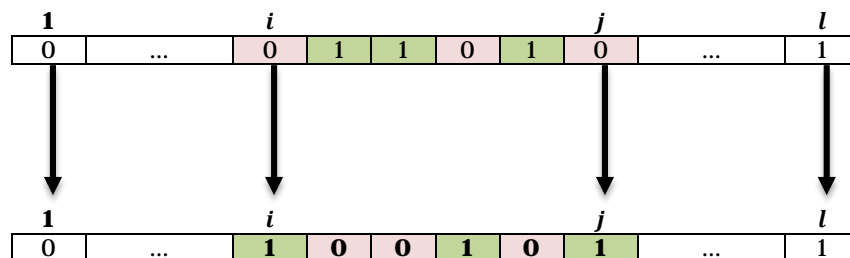
3.2.6.2 Operador de hipermutación

Este operador implementa un mecanismo en el cual cada clon sufre una mutación que es inversamente proporcional a su medida de aptitud. Como cualquier operador clásico de mutación, los cambios en el individuo se aplican modificando bits en la cadena binaria a partir de una probabilidad preestablecida.

3.2.6.3 Operador de hipermacromutación

En el operador de hipermacromutación, el número de cambios aplicados al individuo es independiente a la función objetivo. En este caso, se seleccionan de manera aleatoria dos posiciones en la cadena binaria, y se procede a modificar los bits que se encuentran dentro del rango de esas dos posiciones. La figura 25 ilustra la manera en la que se aplica el operador de hipermacromutación a un individuo codificado como una cadena binaria.

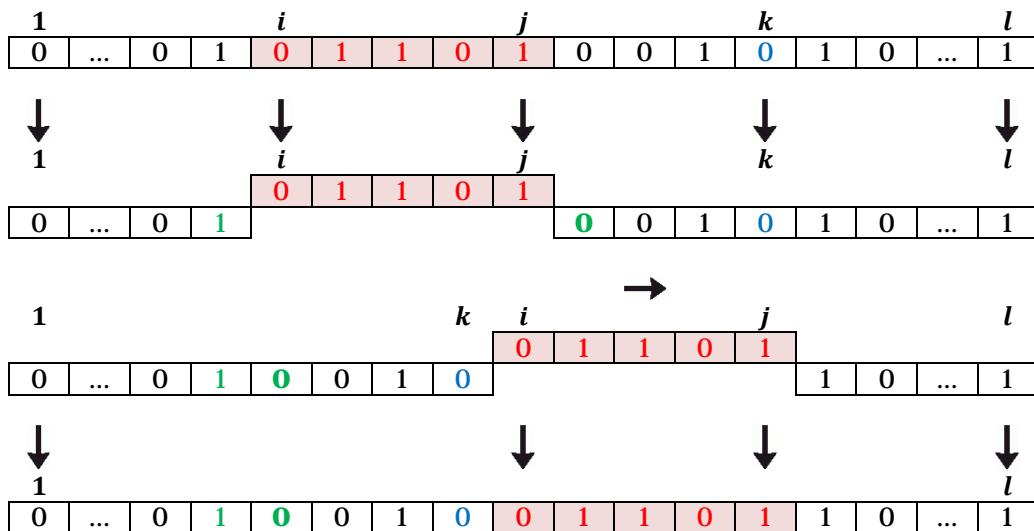
Figura 25. Ejemplo del operador de hipermacromutación aplicado a un genotipo binario.



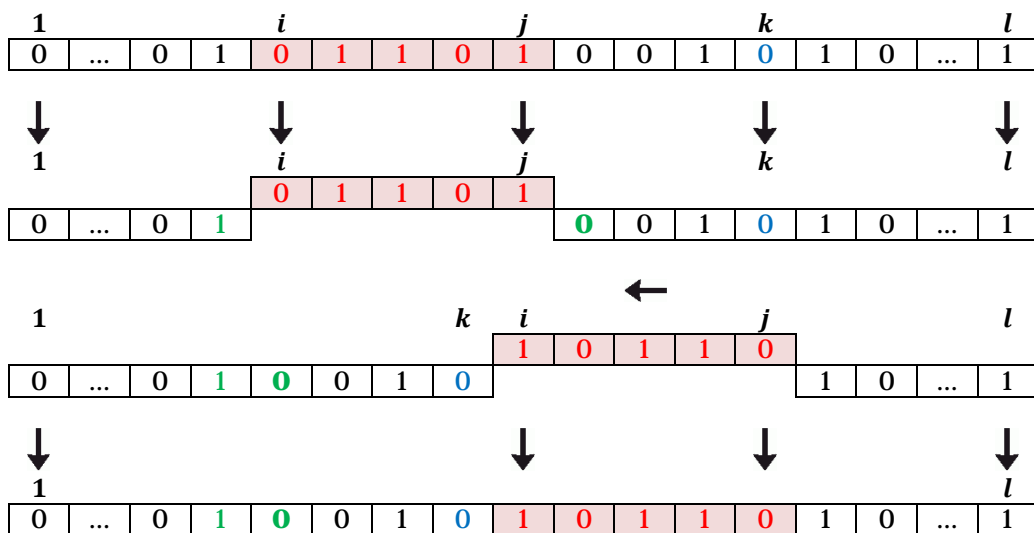
3.2.6.4 Operador de transmutación

En el operador de transmutación se seleccionan de manera aleatoria dos posiciones al interior de la cadena binaria. La subcadena entre los dos puntos seleccionados, es extraída para luego ser insertada en una nueva posición seleccionada también aleatoriamente. El elemento extraído podrá ser insertado en la misma dirección en la que se encontraba inicialmente o de manera inversa. Este tipo de mutación es una analogía a los elementos genéticos conocidos como transposones, que son secuencias de ADN con la capacidad de moverse a diferentes partes del genoma. La figura 26 ilustra la manera en que se aplica el operador de transmutación.

Figura 26. Ejemplo del operador de Transmutación aplicado en un genotipo binario de longitud l . El elemento transponible lo constituyen los bits entre los rangos $[i, j]$. (a) El transposón es insertado luego de la posición k del genotipo (b) El transposón puede también ser insertado de manera reversa.



(a)



(b)

3.2.6.5 Operador de selección

Si luego de aplicar los operadores de mutación, alguno de los clones mutados posee una mejor medida de aptitud que aquel individuo que lo engendró, entonces el clon tomará el lugar de su progenitor en la población.

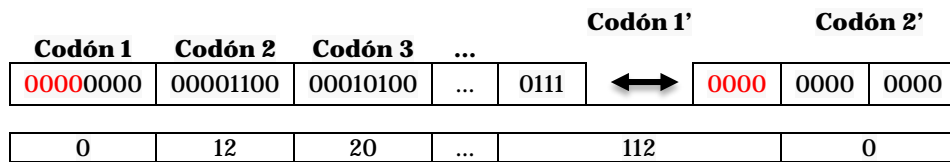
3.2.6.6 Operador de envejecimiento

En caso que un individuo no produzca descendencia que sea capaz de reemplazarlo en la población, envejecerá cada generación. El algoritmo establece un límite de edad para los elementos de la población; cuando se alcance este valor, el individuo morirá y será removido de la población. Las soluciones removidas pasarán a hacer parte de un repositorio denominado subpoblación de memoria. Esta subpoblación de memoria contendrá en efecto a todos los individuos que no lograron evolucionar durante cierto periodo de tiempo, y por lo tanto se asume que han convergido a soluciones óptimas locales.

3.2.6.7 Operador de envolvimiento (*wrapping*)

Cuando el final de una cadena binaria que representa una trayectoria es alcanzado, la cadena completa puede ser reutilizada múltiples veces como si se tratara de una estructura circular. El algoritmo establece un parámetro que determina cuántas veces puede ser reutilizada la cadena binaria. La figura 27 muestra un ejemplo de la manera en que se implementa el operador de envolvimiento.

Figura 27. Ejemplo de la aplicación del operador de envolvimiento. Al alcanzar el final del genotipo, se reutilizan los bits del inicio.



3.2.7 Condición de parada

Al final de cada generación, y luego de aplicar los diversos operadores genéticos, un porcentaje de individuos aleatorios se introduce a la población con el fin de reemplazar a aquellos considerados como los “peores”. El proceso de evolución continúa hasta que se alcanza un número máximo de generaciones definidas a priori, o si el promedio de aptitud toda la población se estabiliza y estanca luego de un número determinado de iteraciones. Lo cual indica que toda la población ha convergido a una solución.

La figura 28 resume el algoritmo utilizado como motor de búsqueda del método de evolución gramatical propuesto para la evolución de trayectorias de plegamiento y predicción de la estructura secundaria de secuencias de ARN.

Figura 28. Pseudocódigo del algoritmo opt-aiNET implementado como motor de búsqueda.

Algoritmo 1: opt-aiNET implementado como motor de búsqueda

P: Población de individuos

M: Subpoblación de memoria

Parámetros:

- **iniP:** tamaño de la población inicial de individuos
- **maxGen:** máximo número de generaciones/iteraciones
- **nc:** número máximo de clones que una célula puede criar
- **pm:** probabilidad de mutación
- **maxAge:** edad máxima que puede alcanzar un individuo
- **et:** umbral de error (Parámetro de convergencia)
- **div:** Porcentaje de nuevos individuos a adicionar en la población
- **maxP:** tamaño máximo de la población

1. Crear la población inicial de individuos de tamaño **iniP**
2. $gen = 0$
 - MIENTRAS** el criterio de parada no se cumpla **HACER**
 3. Evalúe la aptitud de cada elemento de la población
 4. Clone cada elemento de la población según su valor de aptitud
 5. Aplique los operadores de mutación a la población de clones
 6. Aplique el operador de envolvimiento a cada clon mutado
 7. **PARA** cada individuo de la población **HACER**
 8. Seleccionar el clon mutado con mejor aptitud
 9. **SI** la aptitud del clon mutado es mejor que la de su progenitor **ENTONCES**
 10. El clon mutado reemplazará a su padre en la población
 11. Inicialice la edad del clon en 0
 12. **SI NO**
 13. La edad del progenitor aumentará una unidad
 14. **FIN SI**
 15. **SI** la edad del progenitor es igual a **maxAge** **ENTONCES**
 16. El individuo es removido de la población **P** y es enviado a la subpoblación de memoria **M**
 17. **FIN SI**
 18. **FIN PARA**
 19. Calcule la aptitud promedio de toda la población
 20. **SI** la aptitud promedio no se ha estancado,
 21. retorne al inicio del ciclo
 22. **SI NO**
 23. Suprima individuos similares.
 24. Introduzca un porcentaje **div** de nuevos individuos a la población
 25. Suprima los peores individuos de la población de tal forma que el tamaño de la población no supere **maxP**.
 26. **FIN SI**
 27. $gen = gen + 1$
 28. **FIN**

4. Pruebas y resultados

Para evaluar el desempeño del algoritmo propuesto, se llevó a cabo un conjunto de pruebas utilizando secuencias para las cuales la estructura secundaria ya se encuentra determinada. Cada estructura obtenida luego del proceso de predicciones fue comparada con aquellas generadas por el servidor Mfold, descrito en (M. Zuker, 2003).

Las secuencias utilizadas en esta etapa se organizaron en tres grupos según sus características. El objetivo de cada prueba consistía en predecir, a partir de la secuencia de nucleótidos, la estructura secundaria con la menor energía libre posible y una trayectoria de plegamiento que condujera a dicha estructura.

El algoritmo fue implementado en el lenguaje MATLAB y ejecutado en la versión R2013b de dicha herramienta.

Las pruebas se realizaron en una estación de trabajo con un procesador Intel Xeon de seis núcleos a 2.6 GHz, 32 GB de memoria RAM y con sistema operativo Ubuntu.

Los parámetros del algoritmo evolutivo fueron fijados por defecto para todas las pruebas realizadas y el conjunto de estos se presenta en la tabla 6. Para los cálculos de energía, fueron utilizados los parámetros del modelo termodinámico definido en (Douglas H Turner & Mathews, 2010)

Tabla 6. Conjunto de parámetros establecidos para las pruebas realizadas

Parámetro	Valor
Número inicial de individuos	10
Tamaño máximo de la población	30
Longitud de un individuo	[1000, 1500]
Máximo número de generaciones	40
Número máximo de clones por individuo	5
Probabilidad de mutación	40%
Porcentaje de nuevos individuos a ser insertados	40%
Edad máxima que puede alcanzar un individuo	10
Parámetro de envolvimiento (wrapping)	4
Parámetro de convergencia	0.01

4.1 Primer escenario: Riboswitches

Para el primer escenario de evaluación se seleccionaron 15 secuencias utilizadas en un trabajo previo en el que se evaluaba el desempeño de un algoritmo genético, denominado RNAEAPath, diseñado para la determinación de trayectorias de plegamiento (Li & Zhang, 2012). Todas las secuencias seleccionadas corresponden a un tipo de ARN denominados Riboswitches. Las características de cada secuencia se presentan en la tabla 7.

Tabla 7. Características de las secuencias seleccionadas para el primer escenario de pruebas

Instancia	ID	nt	pb	Tallos	ΔG [kcal/mol]
1	rb1	148	44	6	-55.70
2	rb2	113	29	5	-24.80
3	rb3	141	50	11	-51.40
4	rb4	146	47	10	-52.90
5	rb5	201	56	9	-65.60
6	SplicedLeaderAB	56	17	5	-10.70
7	attenuator	73	18	4	-21.20
8	s15	74	19	4	-17.90
9	sbox leader	247	73	12	-84.60
10	thiM leader	165	45	11	-53.70
11	ms2	73	24	4	-32.10
12	HDV	153	47	10	-72.10
13	dsrA	85	30	4	-32.30
14	amv	145	48	7	-52.00
15	alpha operon	130	37	8	-37.10

Un total de 10 ejecuciones fueron realizadas para cada secuencia. Los porcentajes de ocurrencia de las estructuras obtenidas con los menores y mayores valores de energía libre se encuentran resumidos en la Tabla 8. Se estimó el porcentaje de pares de bases predichas correctamente en cada caso.

Tabla 8. Resultados obtenidos en el primer escenario de pruebas utilizando Riboswitches.

ID	MFE <i>Mfold</i> ΔG [kcal/mol]	Predicción				
		Mejor caso		Peor caso		
		ΔG	f_i (%)	pb (%)	ΔG	pb (%)
rb1	-55.70	-55.70	100.0	100.0	-	-
rb2	-24.80	-24.80	100.0	100.0	-	-
rb3	-51.40	-51.40	40.0	100.0	-45.70	32.0
rb4	-52.90	-52.90	100.0	100.0	-	-
rb5	-65.60	-65.60	70.0	100.0	-60.10	91.0
SplicedLeaderAB	-10.70	-10.70	100.0	100.0	-	-
attenuator	-21.20	-21.20	100.0	100.0	-	-
s15	-17.90	-17.90	100.0	100.0	-	-
sbox leader	-84.60	-82.40	40.0	52.1	-65.40	50.7
thiM leader	-53.70	-53.30	25.0	100.0	-51.90	39.7
ms2	-32.10	-32.10	100.0	100.0	-	-
HDV	-72.10	-69.30	40.0	95.7	-68.50	76.6
dsrA	-32.30	-32.30	100.0	100.0	-	-
amv	-52.00	-51.70	100.0	93.8	-	-
alpha operon	-37.10	-37.10	50.0	100.0	-36.80	94.6

De los resultados obtenidos se aprecia que en 12 de las 15 instancias el algoritmo encuentra la estructura con la mínima energía. Adicionalmente se evidencia que en 14 de los 15 casos, el porcentaje de pares de bases predichos de manera correcta se encuentra por encima del 90%, lo cual es un buen indicador de que la simulación del proceso de plegamiento se realiza de manera correcta.

Para el caso de la secuencia identificada como **sbox leader** no fue posible igualar el mejor resultado obtenido por *Mfold*, debido a que la estructura predicha por este último contiene tallos conformados solo por dos pares de bases, lo cual no cumple con el planteamiento del modelo propuesto en el cual se define que todo tallo al interior de una estructura secundaria debe contener como mínimo tres pares de bases para considerarse estable. En ese sentido, el algoritmo queda atrapado en óptimos locales al no poder construir todos los motivos estructurales presentes en la estructura secundaria con el menor valor de energía. La mejor estructura obtenida para esta secuencia presenta un valor de energía libre de -82.40 kcal/mol que corresponde a la tercera de cinco estructuras que logra predecir *Mfold*. Los valores de energía de estas estructuras generadas por *Mfold* son presentadas en la figura 29.

Figura 29. Resultados generados por Mfold para la secuencia identificada como **sbox leader**.

<p>♦ Structure 1 : Initial $\Delta G = -84.60$ kcal/mol, (Thermodynamic Details). Different file formats: PostScript, pdf, png, jpg, .ct file, Vienna, RNAML, RnaViz ct, Mac ct, RNAdraw, XRNA ss.</p>
<p>♦ Structure 2 : Initial $\Delta G = -83.10$ kcal/mol, (Thermodynamic Details). Different file formats: PostScript, pdf, png, jpg, .ct file, Vienna, RNAML, RnaViz ct, Mac ct, RNAdraw, XRNA ss.</p>
<p>♦ Structure 3 : Initial $\Delta G = -82.40$ kcal/mol, (Thermodynamic Details). Different file formats: PostScript, pdf, png, jpg, .ct file, Vienna, RNAML, RnaViz ct, Mac ct, RNAdraw, XRNA ss.</p>
<p>♦ Structure 4 : Initial $\Delta G = -80.70$ kcal/mol, (Thermodynamic Details). Different file formats: PostScript, pdf, png, jpg, .ct file, Vienna, RNAML, RnaViz ct, Mac ct, RNAdraw, XRNA ss.</p>
<p>♦ Structure 5 : Initial $\Delta G = -80.50$ kcal/mol, (Thermodynamic Details). Different file formats: PostScript, pdf, png, jpg, .ct file, Vienna, RNAML, RnaViz ct, Mac ct, RNAdraw, XRNA ss.</p>

Fuente: mfold Web Server (M. Zuker, 2003)

Una de las trayectorias de plegamiento ilustrando la formación de los tallos para la secuencia identificada como **SplicedLeaderAB** se muestra en la Tabla 9. Los valores de energía para cada estructura son igualmente presentados.

Tabla 9. Un ejemplo de una de las trayectorias de plegamiento obtenidas al aplicar el algoritmo propuesto a la secuencia identificada como SplicedLeaderAB.

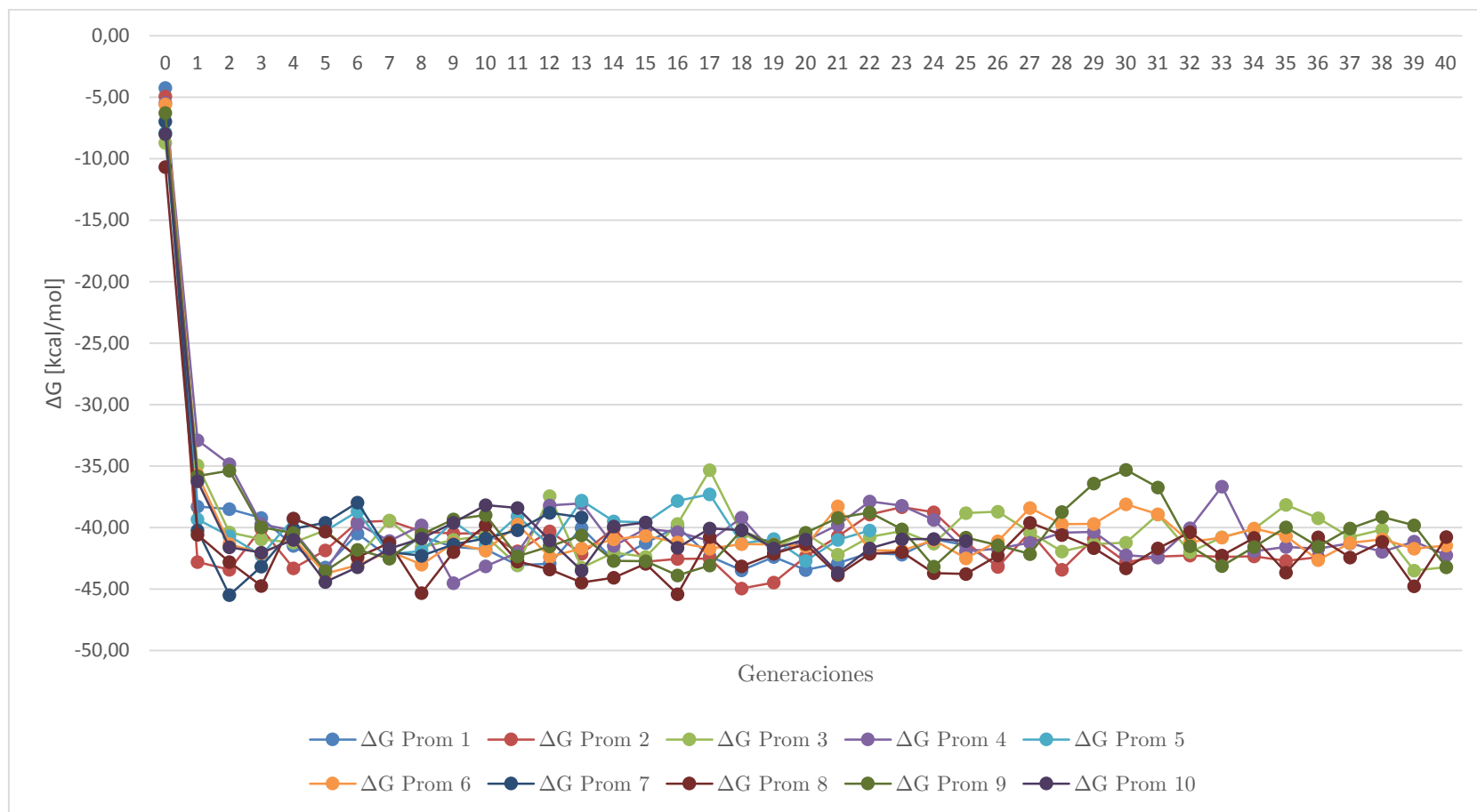
Etapa	Estructura secundaria	ΔG
0	[:aacuaaaacaauuuugaagaacaguuuucuguaacuucuaugguauguagagacuuc]	0.00
1	[:aacuaaaacaauuuu[WYZZY:aacaguuuucugua]uugguauguagagacuuc]	-2.10
2	[:aacuaaa[ZXZ:auuuu[WYZZY:aacaguuuucugua]uuggua]agagacuuc]	-2.80
3	[:aacuaaa[ZXZ:auuuu[WYZZY:a[ZXZY:uuu]a]uuggua]agagacuuc]	-8.10
4	[:aacuaaa[ZXZ:[ZNN:uu[WYZZY:a[ZXZY:uuu]a]uu]a]agagacuuc]	-10.10
5	[:aa[XW:aaa[ZXZ:[ZNN:uu[WYZZY:a[ZXZY:uuu]a]uu]a]ag]acuuc]	-10.70

El ejemplo de la trayectoria de plegamiento obtenida muestra que el modelo gramatical diseñado para simular las operaciones de plegamiento, opera de manera correcta; guiando en cada etapa de la transformación de las estructuras intermedias hacia conformaciones con valores de energía menores.

En el anexo E se encuentran presentados los mejores plegamientos para cada una de las 15 estructuras utilizadas en esta prueba.

La figura 30 ilustra la tendencia decreciente de la energía libre promedio de la población de individuos para 10 ejecuciones de la secuencia identificada como **amv**. El gráfico evidencia que el algoritmo tiende a evolucionar las trayectorias con el fin de hacer converger las estructuras finales a valores mínimos de energía libre.

Figura 30. Evolución de la energía libre promedio de toda la población de individuos para el caso de 10 ejecuciones.



4.2 Segundo escenario: ARN de transferencia (ARNt)

El ARN de transferencia es un tipo de ácido nucleico encargado de transportar los aminoácidos. Este tipo de moléculas juegan un papel crucial en el proceso de síntesis de proteínas. Existe una molécula de ARNt para cada uno de los 21 aminoácidos que forman parte de las proteínas.

Una de las características del ARNt es que la estructura estable que adopta en su estado natural no suele corresponder a aquella con la menor energía libre. La estructura secundaria de este tipo de moléculas suele asemejarse a un trébol de 4 hojas (*cloverleaf structure*). Un ejemplo de lo anterior se ilustra en la figura 31, que presenta la estructura secundaria de un ARNt correspondiente al aminoácido Fenilalanina.

Para este segundo escenario de pruebas se seleccionaron de manera aleatoria 21 secuencias de ARNt de la especie Homo Sapiens, Cada una correspondiente a un aminoácido distinto. Las secuencias fueron obtenidas en la Base de datos de ARN de transferencia tRNAdb 2009 (Jühling et al., 2009).

Aunque naturalmente los ARNt no adoptan la conformación mínima de energía libre, las pruebas se orientaron a encontrar aquella estructura con el mínimo valor de energía, así como las trayectorias que conducen a dichas estructuras.

Para el segundo escenario se realizaron 100 ejecuciones para cada secuencia. Los resultados obtenidos se resumen en la Tabla 10.

Figura 31. Estructura secundaria de un ARNt - Phe.

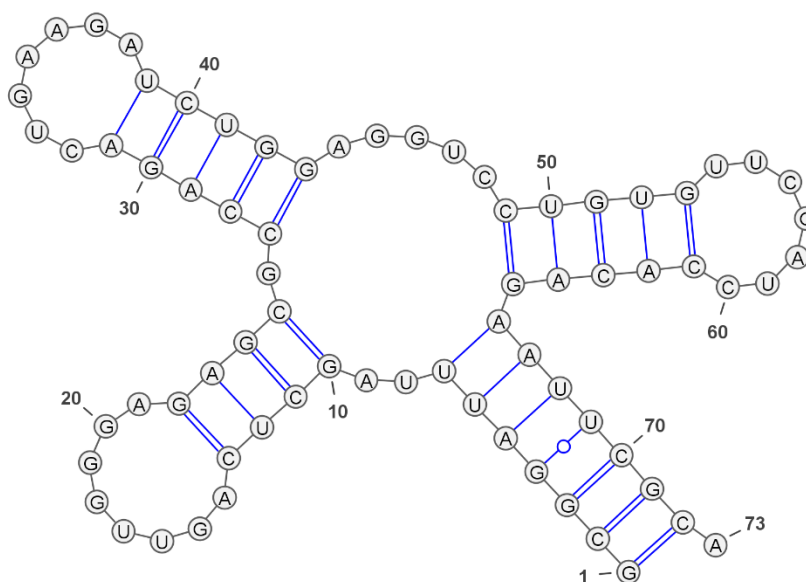


Tabla 10. Resultados obtenidos en el primer escenario de pruebas utilizando un conjunto de secuencias de ARNt.

Aminoácido	ARNt				Mfe Mfold		Predicción	
	ID	nt	bp	Tallos	ΔG [kcal/mol]	ΔG	f_i (%)	bp (%)
Alanina	Ala	73	18	5	-22.60	-22.60	100.0	100.0
Arginina	Arg	73	22	4	-27.20	-27.20	100.0	100.0
Asparagina	Asn	74	24	5	-26.10	-26.10	100.0	100.0
Aspartato	Asp	72	20	4	-23.00	-23.00	100.0	100.0
Cisteína	Cys	72	23	5	-27.20	-27.20	100.0	100.0
Glutamina	Gln	74	20	4	-18.60	-18.60	100.0	100.0
Ácido glutámico	Glu	69	21	4	-24.60	-24.60	100.0	100.0
Glicina	Gly	71	19	5	-25.50	-25.50	100.0	100.0
Histidina	His	72	22	5	-20.30	-19.00	100.0	86.4
Isoleucina	Ile	74	24	6	-23.40	-23.40	100.0	100.0
Leucina	Leu	82	28	5	-27.70	-27.70	100.0	100.0
Lisina	Lys	73	18	4	-17.60	-17.60	100.0	100.0
Metionina	Met	75	24	5	-25.90	-25.90	100.0	100.0
Fenilalanina	Phe	76	23	5	-17.20	-17.20	100.0	100.0
Prolina	Pro	72	21	4	-30.80	-30.80	100.0	100.0
Serina	Ser	82	21	5	-18.80	-18.80	100.0	100.0
Treonina	Thr	74	19	5	-22.30	-22.30	100.0	100.0

Triptófano	Trp	74	21	5	-26.40	-26.40	100.0	100.0
Tirosina	Tyr	93	26	5	-29.40	-29.40	100.0	100.0
Valina	Val	73	15	4	-10.80	-10.80	100.0	100.0
Selenocisteína	Sec	74	20	4	-20.60	-20.60	100.0	100.0

De los resultados obtenidos se puede evidenciar que en 20 de las 21 de las instancias, el algoritmo encuentra la estructura secundaria con el menor valor de energía en el 100% de las ejecuciones. En el caso del ARNt del aminoácido Histidina el algoritmo no logra predecir la estructura óptima; sin embargo, la estructura encontrada comparte el 86.4% de los pares de bases con la estructura con el valor de energía mínimo. Un análisis detallado de este caso en particular arrojó que Mfold genera un bucle en una región de bases complementarias forzando que pares de bases del tipo G-U al interior de un tallo no se formen. Según explicación del autor esto acontece debido a que en los casos en donde se encuentran dos pares de bases del tipo GU apilados, el cálculo de este elemento estructural se realiza utilizando un conjunto de parámetros especiales que alteran el valor de la energía. Una ilustración de la región en donde se presenta la divergencia entre la estructura predicha y aquella encontrada por Mfold se muestra en la figura 32.

En ese orden de ideas, debido a que la gramática que simula la formación de los tallos obliga a la formación de todos los pares de bases en regiones complementarias se puede concluir que el algoritmo propuesto encuentra las estructuras energéticamente más estables en todas las instancias evaluadas.

Figura 32. Comparación del resultado obtenido para la secuencia de ARNt – His.

Estructura secundaria (Mfold)

GCAGUGACUGUAUAGUGGUUAGCACUCUGUGUUGUGGCCACAGCAACCAUGGUUCAAAUCUGAGUCAUGACA
 ((((. . .))) . . . (((((((((((((. . .)))) . . .)))) ((((. . . ((((.))))

Estructura secundaria

ΔG

Mfold	[: [MXZY:uga] aua [YWYYN: [NZMXZX:ucu] ug] agcaac [XZWY:gu [WXZ:aauc] gu] aca]	-20.30
Predicción	[: [MXZY:uga] aua [YWYYN: [NZMXZX:ucu] ug] agcaac [XZWYMNWXZ:aauc] aca]	-19.00

En el anexo E se encuentran presentados los mejores plegamientos para cada una de las 15 estructuras utilizadas en esta prueba.

4.3 Tercer escenario: Dominios de receptores de Dopamina

En este tercer escenario se realizaron pruebas con dominios extraídos de secuencias consenso de receptores de dopamina. Un dominio se define como una región conservada dentro de una secuencia. Por su parte, una secuencia consenso se obtiene a partir de un alineamiento de múltiples secuencias, y busca especificar cuáles son los elementos comunes en regiones de las secuencias alineadas.

En este escenario, se obtuvieron estructuras consenso correspondientes a los 5 tipos de receptores de dopamina a partir de 230 secuencias de ARN mensajero (ARNm) y proteínas obtenidas del GenBank Database (Benson, Karsch-Mizrachi, Lipman, Ostell, & Wheeler, 2006). Posteriormente 7 Regiones conservadas fueron identificadas y extraídas de cada secuencia.

Cabe señalar que la predicción de estructuras de segmentos de secuencia no tiene sentido biológico debido a que se está ignorando las interacciones que los nucleótidos de la sección extraída puedan tener con el resto de la molécula. Sin embargo el objetivo de la prueba consistió en evaluar si el algoritmo propuesto puede hallar estructuras con el mínimo valor de energía aunque no se trate de secuencias completas.

Para esta prueba se realizaron 10 ejecuciones con cada dominio y se registraron las frecuencias de ocurrencia de las estructuras con menor y mayor energía libre. Los resultados para cada dominio se resumen en las Tablas 11 a 17

Tabla 11. Resultados obtenidos para el primer dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[:cu[YX:u[XZ:u[XX:u[YN:ccacg[XN:ccugg]aacac]u]uc]u]ugccguc]	-6.00	
	Mejor	[:cu[YX:u[XZN:ccugu[XXZ:c[YX:[WXX:ugg]acac]]ucu]cu]cguc]	-5.90	1
	Peor	[:cugcucauccugu[XXZ:cgcucc][YY:aa[XZXM:cugguc]cug]guc]	-4.60	3
DR5	Mfold	[:[XWYX:u[XZ:u[XZNX:[NY:ga[XXX:ugcu]caa]ugcu]]c]ccauc]	-12.60	
	Mejor	[:[XWYX:u[XZ:u[XZNX:[NY:ga[XXX:ugcu]caa]ugcu]]c]ccauc]	-12.60	9
	Peor	[:c[WYXNXZ:u[XZ:ucuggaccc]c]ac[YNYX:uggugu]agccauc]	-12.40	1
DR2	Mfold	[:c[WYXN:caucuu[XY:ucau]ucuuc]ac[YWY:c[WYYW:g[WYXZ:ug[YXX:gug[WX:ccgc]gaa]gc]g]c]caac]	-23.80	
	Mejor	[:c[WYXN:caucuu[XY:ucau]ucuuc]ac[YWY:c[WYYW:g[WYXZ:ug[YXX:gug[WX:ccgc]gaa]gc]g]c]caac]	-23.80	10
	Peor	-	-	-
DR3	Mfold	[:[MXMX:u[XZNX:cu[YMXXZW:c[YN:cuucg]a]u]u]ggcugug]	-18.60	
	Mejor	[:[MXMX:u[XZNX:cu[YMXXZW:c[YN:cuucg]a]u]u]ggcugug]	-18.60	10
	Peor	-	-	-
DR4	Mfold	[:cu[YXWXZN:[XYM:c[YNYM:u[YWWX:gcggg]]]ugugc]gug]	-16.00	
	Mejor	[:cu[YXWXZN:[XYM:c[YNYM:u[YWWX:gcggg]]]ugugc]gug]	-16.00	10
	Peor	-	-	-

Tabla 12. Resultados obtenidos para el segundo dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[:uuu[YNXZN:c[WXX:uu[YYX:u[MWY:[NX:[ZYZN:cu[XN:uggu]cu]u]u]]cu]aagca]u]	-16.40	
	Mejor	[:uuu[YNXZN:c[WXX:uu[YYX:u[MWY:[NX:[ZYZN:cu[XN:uggu]cu]u]u]]cu]aagca]u]	-16.40	6
	Peor	[:uuu[YNXZ:ucuccu]ugugucagaucucuug[MWYMXN:gucuu][YXX:[XWY:gaaag]u]u]	-15.00	4
DR5	Mfold	[:[WNXZ:uc[YNMW:cu[X:ug[YX:c[YNYNX:[XYZ:ccucu]u]u]ug]uc]cc]a[YYX:gguc]]	-22.90	
	Mejor	[:[WNXZ:uc[YNMW:cu[XNMYX:c[YNYNX:[XYZ:ccucu]u]u]uc]cc]a[YYX:gguc]]	-22.60	10
	Peor	-	-	-
DR2	Mfold	[:cugaucgu[XZMXXN:c[YXY:g[WYYXX:g[ZXX:uccu]]]u[YY:ucaug]cu]ucuaccug]	-25.10	
	Mejor	[:cugaucgu[XZMXXN:c[YX:g[WYYXX:g[ZXX:uccu]]]u[YY:ucaug]cu]ucuaccug]	-25.10	10
	Peor	-	-	-
DR3	Mfold	[:cu[MYN:[YYW:ga[YXX:uggcugu]ag]u][YYWM:[MXXZXXN:u[YY:ugauc]u]g]ug]	-26.50	
	Mejor	[:cu[MYN:[YYW:ga[YXX:uggcugu]ag]u][YYWM:[MXXZXXN:u[YY:ugauc]u]g]ug]	-26.50	10
	Peor	-	-	-
DR4	Mfold	[:uucau[XY:u[YZYX:cu[YYXMYX:c[MXXM:accuccuccucgcucc]]]uu]ucuacucc]	-18.80	
	Mejor	[:uucau[XY:u[YZYX:cu[YYXMYX:c[MXXM:accuccuccucgcucc]]]uu]ucuacucc]	-18.80	10
	Peor	-	-	-

Tabla 13. Resultados obtenidos para el tercer dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[: [YZ : [XZNZWYMWY : gccuuuuga] c] c [ZXN : gc [YWXXZN : c [XW : ca [ZX : cucugu] gauc] c] a] ac]	-20.10	
	Mejor	[: [YZ : [XZNZWYMWY : gccuuuuga] c] c [ZXN : gc [YWXXZN : c [XW : ca [ZX : cucugu] gauc] c] a] ac]	-20.10	10
	Peor	-	-	-
DR5	Mfold	[: [YZXYN : cu [MYYW : [YYX : cuuc [YZ : [XZN : cau] c] cacc] uccauccuga] gu] auc [ZYXY : uggac] ac]	-17.80	
	Mejor	[: [YZXYN : cu [MYYW : [YYX : cuuc [YZ : [XZN : cau] c] cacc] uccauccuga] gu] auc [ZYXY : uggac] ac]	-17.80	10
	Peor	-	-	-
DR2	Mfold	[: [YZXX : ucuu] ac [WX : u [YYZWYN : ca [WY : a [WY : ugca] g] a] u] [ZXXWYW : [MWYX : cauca] cg] ac]	-23.30	
	Mejor	[: [YZXX : ucuu] ac [WX : u [YYZWYN : ca [WY : a [WY : ugca] g] a] u] [ZXXWYW : [MWYX : cauca] cg] ac]	-23.30	10
	Peor	-	-	-
DR3	Mfold	[: gau [YW : u [WNWYWX : acccug [YZWYN : ca [WYZWY : u [YNZXZY : c [XZY : cauc] aaccu]]]] u]]	-21.60	
	Mejor	[: gau [YW : u [WNWYWX : acccug [YZWYN : ca [WYZWY : u [YNZXZY : c [XZY : cauc] aaccu]]]] u]]	-21.60	10
	Peor	-	-	-
DR4	Mfold	[: gac [YXX : cucau] c [ZWYYZ : u [YN : ca [WYX : ugu] cc] c] cuuc [ZXXXWYW : [MXYX : cauca] gg] c]	-22.80	
	Mejor	[: gac [YXX : cucau] c [ZWYYZ : u [YN : ca [WYX : ugu] cc] c] cuuc [ZXXXWYW : [MXYX : cauca] gg] c]	-22.80	10
	Peor	[: [YZ : [XZNZWYMWY : gccuuuuga] c] c [ZXN : gc [YWXXZN : c [XW : ca [ZX : cucugu] gauc] c] a] ac]	-20.10	

Tabla 14. Resultados obtenidos para el cuarto dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[: aucc[WYZ:uc[Z Y:c[MWY:[YXZNMYZX:cuu]u]]ccu]uc[XXZYN:gcagcuca]]	-17.60	
	Mejor	[: aucc[WYZ:uc[Z Y:cgu[MYXZNMYZX:cuu]cau]ccu]uc[XXZYN:gcagcuca]]	-16.30	10
	Peor	-	-	-
DR5	Mfold	[: gucauggu[XMY:cc[WY:gc[XWYYZXX:uuguccauccucaucuccuaucaucc]cu]a]g]	-14.30	
	Mejor	[: gucauggu[XMY:cc[WY:gc[XWYYZXX:uuguccauccucaucuccuaucaucc]cu]a]g]	-14.30	10
	Peor	-	-	-
DR2	Mfold	[: a[XXY:uc[ZNYZW:cgcc]c[WYYYN:ccuguccuucaccaucuccu]cugcucu]acuc]	-8.10	
	Mejor	[: a[XXY:uc[ZNYZW:cgcc]c[WYYYN:ccuguccuucaccaucuccu]cugcucu]acuc]	-8.10	10
	Peor	-	-	-
DR3	Mfold	[: [YXX:cu[XZ:u[YZ:u[XZXZYX:cgucu[MYYN:acug]uu]]c]cccucuccucuuu]uuc]	-14.80	
	Mejor	[: gccucau[YZ:u[XZXZYX:cgucu[MYYN:acug]uu]]cu[YXX:cucuccucuuu]uuc]	-14.80	10
	Peor	-	-	-
DR4	Mfold	[: cu[YXNYX:u[XZWXMX:c[YX:c[ZXYN:g[MXNYX:uguccgcu]g]c]cc]u]cuc]	-34.00	
	Mejor	[: cu[YXNYX:u[XZWXMX:c[YX:c[ZXYN:g[MXNYX:uguccgcu]g]c]cc]u]cuc]	-34.00	10
	Peor	-	-	-

Tabla 15. Resultados obtenidos para el quinto dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[:uauGCCAUcUCAUCCUCCUAUAAGCUUUUACAUCUU[YWYMX:c[ZW:caug]u]CUAC]	-4.70	
	Mejor	[:uauGCCAUcUCAUCCUCCUAUAAGCUUUUACAUCUU[YWYMX:c[ZW:caug]u]CUAC]	-4.70	10
	Peor	-	-	-
DR5	Mfold	[:uac[YXXZNX:uccuccuc[YXW:cauc]uucuaCAUccc]ca[WXZN:gauC]ccuac]	-9.90	
	Mejor	[:uac[YXXZNX:uccuccuc[YXW:cauc]uucuaCAUccc]ca[WXZN:gauC]ccuac]	-9.90	10
	Peor	-	-	-
DR2	Mfold	[:uu[XYWMY:ucuaCuccuccaUcGucuccuu]ugccCUcAUcGucac[XX:ugcu]ucuaC]	-3.20	
	Mejor	[:uu[XYWMY:ucuaCuccuccaUcGucuccuu]ugccCUcAUcGucac[XX:ugcu]ucuaC]	-3.20	10
	Peor	-	-	-
DR3	Mfold	[:uuugUCAUCU[ZXWXN:uc[Zy:u[YYN:guccuCUaccu]]uu][YZXN:guccu]uau]	-11.00	
	Mejor	[:uuugUCAUCU[ZXWXN:uc[Zy:u[YYN:guccuCUaccu]]uu][YZXN:guccu]uau]	-11.00	10
	Peor	-	-	-
DR4	Mfold	[:uac[Y:ug[YN:CUacu[XY:uc[XM:ucU[YX:uccuCUuccu]cc]cc]CUcau]ug]ugcucuaC]	-4.80	
	Mejor	[:uac[YNMYN:CUacu[XY:uc[XM:ucU[YX:uccuCUuccu]cc]cc]CUcau]ugcucuaC]	-4.70	10
	Peor	-	-	-

Tabla 16. Resultados obtenidos para el sexto dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[:cuguc[YMWYZ:ucau[MMYYN: [YN:uu[YN:gugcu]ug]u]cu]uug]	-11.60	
	Mejor	[:cuguc[YMWYZ:ucau[MMYYN: [YN:uu[YN:gugcu]ug]u]cu]uug]	-11.60	10
	Peor	-	-	-
DR5	Mfold	[:cuguc[YMWYZ:ucau[MMYYN: [YN:uu[YN:gugcu]ug]u]cu]uug]	-11.60	
	Mejor	[:cuguc[YMWYZ:ucau[MMYYN: [YN:uu[YN:gugcu]ug]u]cu]uug]	-11.60	10
	Peor	-	-	-
DR2	Mfold	[:cugccau[YN:uc[NXMYX: [MWY:uuc]cu]cu]ccuucuaucacc]	-6.20	
	Mejor	[:cugccau[YN:uc[NXMYXM: [WY:uuc]cu]cu]ccuucuaucacc]	-5.70	10
	Peor	-	-	-
DR3	Mfold	[:cu[YY:c[XZ:uugucuu[MYYYX:cu[NXZ:uugucugc]cu]ucu]a]]	-9.70	
	Mejor	[:cu[MYX:cau]u[MYYYX:cu[NXZ:uugucugc]cu]ucu]ugacc]	-9.00	10
	Peor	-	-	-
DR4	Mfold	[:cuu[XXNYN:u[YWY:c[WY:ga[XMX:uuucucgu]ca]u]]cgcugug]	-16.30	
	Mejor	[:cuu[XXNYN:u[YWY:c[WY:ga[XMX:uuucucgu]ca]u]]cgcugug]	-16.30	10
	Peor	-	-	-

Tabla 17. Resultados obtenidos para el séptimo dominio de los receptores de dopamina.

Receptor		Estructuras Secundarias	ΔG	f_i
DR1	Mfold	[:u[YYWW:ug[MY:ug[YY:cuaauu]uc]ug]ccaucuuuuugcc]	-12.20	
	Mejor	[:u[YYWWN:[MY:u[YYM:cuaau]u]u]ccaucuuuuugcc]	-11.70	10
	Peor	-	-	-
DR5	Mfold	[:u[YYWW:ug[MY:ug[YY:cuaauu]uc]ug]ccaucuuuuugcc]	-12.20	
	Mejor	[:u[YYWWN:[MY:u[YYM:cuaau]u]u]ccaucuuuuugcc]	-11.70	10
	Peor	-	-	-
DR2	Mfold	[:ug[YXWY:[YM:cuaug]aa]gcc[MWYZ:acccca]cuacacc]	-4.60	
	Mejor	[:ug[YXWY:ggcuaugucua]gcc[MWYZ:acccca]cuacacc]	-5.80	10
	Peor	-	-	-
DR3	Mfold	[:u[YY:cu[YYYX:uacgugaacagc]ucaa]ccgugaucuaauacc]	-6.50	
	Mejor	[:u[YY:c[WYMY:cua[XYN:gaaca]cc]a]ccgugaucuaauacc]	-6.50	7
	Peor	[:u[YY:cu[YYYX:uacgugaacagc]ucaa]ccgugaucuaauacc]	-6.50	3
DR4	Mfold	[:[WYY:cu[YYYX:uacgucaacagc]ucaacc]ucaucuaacacc]	-6.60	
	Mejor	[:[WYY:cu[YYYX:uacgucaacagc]ucaacc]ucaucuaacacc]	-6.60	10
	Peor	-	-	-

Analizando los resultados obtenidos luego de ejecutar las pruebas en el tercer escenario. Se evidencia que el algoritmo en el 100% de los dominios, predice el valor de energía mínimo asociado a la molécula y a su vez simula el proceso que atraviesa ésta para llegar a estado estable. Para los casos de los dominios tres, cuatro, cinco y seis; en la totalidad de las ejecuciones el algoritmo es capaz de igualar el resultado arrojado por Mfold.

5. Conclusiones y recomendaciones

A continuación se presentan las conclusiones del presente trabajo de investigación, así como las recomendaciones para futuros trabajos.

5.1 Conclusiones

El presente trabajo explora una metodología para abordar el problema de predicción de estructuras secundarias de moléculas de ARN desde la perspectiva de la computación evolutiva. Adicionalmente, a partir de la caracterización de los fenómenos presentes en el proceso de plegamiento de los ácidos nucleicos se diseñaron modelos generativos que permiten codificar la información estructural de las moléculas de ARN a la vez que simulan las interacciones físicas y energéticas relacionadas con la conformación de las estructuras secundarias.

La selección de gramáticas como herramienta de modelamiento permitió no solo procesar las moléculas de ARN y su estructura como cadenas de caracteres, sino que hizo posible que los fenómenos biológicos presentes en los procesos de plegamiento fueran susceptibles al análisis computacional, esto último gracias a la correspondencia entre el ensamblaje de estructuras y las reglas de producción de las gramáticas propuestas.

Al asumir que el ARN se encuentra naturalmente en estados termodinámicos de baja energía, es posible plantear la predicción de estructuras como un problema de optimización discreta, a su vez, este último puede ser formulado como un problema de búsqueda que se fundamenta en explorar el espacio de todas las posibles estructuras asociadas a determinada molécula. Debido al tamaño del espacio que aloja estas posibles estructuras, una búsqueda exhaustiva de aquella con el menor valor de energía no es factible, siendo entonces necesario utilizar enfoques heurísticos para llevar a cabo este proceso. En el marco del presente trabajo fue posible implementar una meta-heurística basada en gramáticas con el fin de realizar el proceso de exploración del espacio de todas las estructuras secundarias a la vez que se

minimizaba una función objetivo que asigna valores de energía a las posibles conformaciones que una molécula de ARN puede adoptar.

Los métodos computacionales orientados a la predicción de la estructura secundaria que hacen uso de la energía libre como métrica de estabilidad, requieren integrar modelos termodinámicos simples pero robustos, de tal forma que permitan estimar de manera eficiente las contribuciones energéticas de los elementos estructurales que aparecen durante el plegamiento de una molécula. El algoritmo propuesto no sólo evalúa la estructura final de la molécula, sino también estima las etapas que dan lugar a dicha estructura, brindando información sobre posibles trayectorias de plegamiento gracias a la caracterización de cada una de las estructuras intermedias que surgen a lo largo del proceso de plegamiento de la molécula.

5.2 Recomendaciones

Los problemas de búsqueda y optimización asociados a la predicción de información estructural de macromoléculas suelen acarrear altos costos computacionales. En ese orden de ideas se hace necesario que los futuros desarrollos tengan presentes el desarrollo de soluciones enmarcadas en el área de la computación de alto rendimiento. Se recomienda evaluar la implementación de los algoritmos sobre arquitecturas paralelas y/o distribuidas, con el fin de mejorar su rendimiento.

En el presente trabajo se utilizó la función de energía libre de Gibbs como única métrica de aptitud, en un futuro se recomienda explorar la integración de información adicional que permita integrar información estructural como punto de partida para el proceso de predicción.

La mayoría de los algoritmos de predicción de estructuras secundarias, requieren iniciar a partir de la secuencia completa de nucleótidos. Sin embargo es conocido que las moléculas de ARN inician su proceso de plegamiento durante el proceso de transcripción. Se exhorta explorar el modelamiento a través de gramáticas del proceso de plegamiento de las moléculas mientras estas son sintetizadas.

5.3 Publicaciones

Parte del presente trabajo fue presentado en la 3^a Conferencia Ibero Americana Computación Aplicada 2015, realizada en Florianópolis - Brasil, con artículo titulado:

ALGORITMO EVOLUTIVO BASADO EN GRAMÁTICAS PARA LA PREDICCIÓN DE LA ESTRUCTURA SECUNDARIA DE MOLÉCULAS DE ARN, el cual obtuvo mención de honor por parte del comité de la conferencia, catalogándolo como artículo de la máxima calidad.

Bibliografia

- Achawanantakun, R., Sun, Y., & Takyar, S. S. (2011). NcRNA consensus secondary structure derivation using grammar strings. *Journal of Bioinformatics and Computational Biology*, 09(02), 317–337. <http://doi.org/10.1142/S0219720011005501>
- Achawanantakun, R., Takyar, S. S., & Sun, Y. (2010). Grammar string: a novel ncRNA secondary structure representation. *Lifesciences Society Org*, 2–13. Retrieved from <http://www.lifesciencesociety.org/CSB2010/toc/PDF/2.2010.pdf>
- Agrawal, J., & Agrawal, S. (2015). *Progress in Systems Engineering*. (H. Selvaraj, D. Zydek, & G. Chmaj, Eds.) (Vol. 330). Cham: Springer International Publishing. <http://doi.org/10.1007/978-3-319-08422-0>
- Akutsu, T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104(1-3), 45–62. [http://doi.org/10.1016/S0166-218X\(00\)00186-4](http://doi.org/10.1016/S0166-218X(00)00186-4)
- Amarteifio, S., & O'Neill, M. (2005). Coevolving Antibodies with a Rich Representation of Grammatical Evolution. In *2005 IEEE Congress on Evolutionary Computation* (Vol. 1, pp. 904–911). IEEE. <http://doi.org/10.1109/CEC.2005.1554779>
- Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Wheeler, D. L. (2006). GenBank. *Nucleic Acids Research*, 34(Database issue), D16–20. <http://doi.org/10.1093/nar/gkj157>
- Berman, H. M. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1), 235–242. <http://doi.org/10.1093/nar/28.1.235>
- Bernardino, H., & Barbosa, H. (2009). Artificial Immune Systems for Optimization. *Nature-Inspired Algorithms for Optimisation*, 193, 389–411. http://doi.org/10.1007/978-3-642-00267-0_14
- Bernardino, H. S., & Barbosa, H. J. C. (2009). Grammar-Based Immune Programming for Symbolic Regression. In *Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS '09)* (pp. 274–287). York, UK: Springer-Verlag. http://doi.org/10.1007/978-3-642-03246-2_26

- Bernardino, H. S., & Barbosa, H. J. C. (2010). Grammar-based immune programming. *Natural Computing*, 10(1), 209–241. <http://doi.org/10.1007/s11047-010-9217-x>
- Bernardino, H. S., & Barbosa, H. J. C. (2011a). Grammar-based immune programming. *Natural Computing*, 10(1), 209–241. <http://doi.org/10.1007/s11047-010-9217-x>
- Bernardino, H. S., & Barbosa, H. J. C. (2011b). Inferring systems of ordinary differential equations via grammar-based immune programming. *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6825 LNCS, 198–211. http://doi.org/10.1007/978-3-642-22371-6_19
- Chen, J. H., Le, S. Y., & Maizel, J. V. (2000). Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Research*, 28(4), 991–9. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=102574&tool=pmcentrez&rendertype=abstract>
- Chen, S.-J., & Dill, K. A. (2000). RNA folding energy landscapes. *Proceedings of the National Academy of Sciences*, 97(2), 646–651. <http://doi.org/10.1073/pnas.97.2.646>
- Darty, K., Denise, A., & Ponty, Y. (2009). VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics (Oxford, England)*, 25(15), 1974–5. <http://doi.org/10.1093/bioinformatics/btp250>
- de Castro, L. N., & Timmis, J. (n.d.). An artificial immune network for multimodal function optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (Vol. 1, pp. 699–704). IEEE. <http://doi.org/10.1109/CEC.2002.1007011>
- de Castro, L. N., & Timmis, J. (2002). *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag New York, Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=579419>
- de Franca, F. O., Coelho, G. P., & Von Zuben, F. J. (2010). On the diversity mechanisms of opt-aiNet: A comparative study with fitness sharing. In *IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE. <http://doi.org/10.1109/CEC.2010.5585964>
- de França, F. O., Von Zuben, F. J., & de Castro, L. N. (2005). An artificial immune network for multimodal function optimization on dynamic environments. In *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05* (p. 289). New York, New York, USA: ACM Press. <http://doi.org/10.1145/1068009.1068057>
- Deschenes, A., Wiese, K. C., & Poonian, J. (2004). Comparison of dynamic programming and evolutionary algorithms for RNA secondary structure prediction. In

- Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB '04. Proceedings of the 2004 IEEE Symposium on* (pp. 214–222). IEEE. <http://doi.org/10.1109/CIBCB.2004.1393956>
- Dowell, R. D., & Eddy, S. R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(1), 71. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/15180907>
- Eddy, S. R. (2004). How do RNA folding algorithms work? *Nature Biotechnology*, 22(11), 1457–8. <http://doi.org/10.1038/nbt1104-1457>
- Freier, S. M., Kierzek, R., Jaeger, J. A., Sugimoto, N., Caruthers, M. H., Neilson, T., & Turner, D. H. (1986). Improved free-energy parameters for predictions of RNA duplex stability. *Proceedings of the National Academy of Sciences*, 83(24), 9373–9377. <http://doi.org/10.1073/pnas.83.24.9373>
- Geis, M., & Middendorf, M. (2007). A Particle Swarm Optimizer for Finding Minimum Free Energy RNA Secondary Structures. In *2007 IEEE Swarm Intelligence Symposium* (pp. 1–8). IEEE. <http://doi.org/10.1109/SIS.2007.368019>
- Geis, M., & Middendorf, M. (2011). Particle swarm optimization for finding RNA secondary structures. *International Journal of Intelligent Computing and Cybernetics*, 4(2), 160–186. <http://doi.org/10.1108/17563781111136685>
- Giegerich, R. (2004). Abstract shapes of RNA. *Nucleic Acids Research*, 32(16), 4843–4851. <http://doi.org/10.1093/nar/gkh779>
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. <http://doi.org/10.1023/A:1022602019183>
- Gulyaev, A. P., van Batenburg, F. H., & Pleij, C. W. (1995). The computer simulation of RNA folding pathways using a genetic algorithm. *Journal of Molecular Biology*, 250(1), 37–51. <http://doi.org/10.1006/jmbi.1995.0356>
- Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13), 3429–3431. <http://doi.org/10.1093/nar/gkg599>
- Hugosson, J., Hemberg, E., Brabazon, A., & O'Neill, M. (2010). Genotype representations in grammatical evolution. *Applied Soft Computing Journal*. <http://doi.org/10.1016/j.asoc.2009.05.003>
- Jaeger, J. A., Turner, D. H., & Zuker, M. (1989). Improved predictions of secondary structures for RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 86(20), 7706–10. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=298139&tool=pmcentrez&rendertype=abstract>

- Janssen, S., & Giegerich, R. (2015). The RNA shapes studio. *Bioinformatics (Oxford, England)*, *31*(3), 423–5. <http://doi.org/10.1093/bioinformatics/btu649>
- Jühling, F., Mörl, M., Hartmann, R. K., Sprinzl, M., Stadler, P. F., & Pütz, J. (2009). tRNAdb 2009: compilation of tRNA sequences and tRNA genes. *Nucleic Acids Research*, *37*(Database issue), D159–62. <http://doi.org/10.1093/nar/gkn772>
- Knudsen, B., & Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, *31*(13), 3423–8. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=169020&tool=pmcentrez&rendertype=abstract>
- Krasteva, P. V. (2011). RNA structures. *Nature Methods*, *9*(1), 38–38. <http://doi.org/10.1038/nmeth.1826>
- Li, Y., & Zhang, S. (2012). Predicting folding pathways between RNA conformational structures guided by RNA stacks. *BMC Bioinformatics*, *13*(Suppl 3), S5. <http://doi.org/10.1186/1471-2105-13-S3-S5>
- Luscombe, N. M., Greenbaum, D., & Gerstein, M. (2001). What is bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine*, *40*(4), 346–58. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/11552348>
- Mathews, D. H., Sabina, J., Zuker, M., & Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, *288*(5), 911–40. <http://doi.org/10.1006/jmbi.1999.2700>
- McKinney, B. A., & Tian, D. (2008). Grammatical Immune System Evolution for reverse engineering nonlinear dynamic Bayesian models. *Cancer Informatics*, *6*, 433–47. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2623294&tool=pmcentrez&rendertype=abstract>
- Mcmillan, N. (2006). *RNA Secondary Structure Prediction using Ant Colony Optimisation*. (Master's thesis). University of Edinburgh. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.64.5881>
- Miranda, P. B., & Prudêncio, R. B. (2015). GEFPSO. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15* (pp. 1087–1094). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2739480.2754819>
- Nasuf, A., Bhaskar, A., & Keane, A. J. (2013). Grammatical evolution of shape and its application to structural shape optimisation. *Structural and Multidisciplinary Optimization*, *48*(1), 187–199. <http://doi.org/10.1007/s00158-013-0890-0>
- Nebel, M. E., & Scheid, A. (2011). Evaluation of a sophisticated SCFG design for RNA

- secondary structure prediction. *Theory in Biosciences = Theorie in Den Biowissenschaften*, 130(4), 313–36. <http://doi.org/10.1007/s12064-011-0139-7>
- Neethling, M., & Engelbrecht, A. P. (2006). Determining RNA Secondary Structure using Set-based Particle Swarm Optimization. In *2006 IEEE International Conference on Evolutionary Computation* (pp. 1670–1677). IEEE. <http://doi.org/10.1109/CEC.2006.1688509>
- Nussinov, R., & Jacobson, A. B. (1980). Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 77(11), 6309–13. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=350273&tool=pmcentre&rendertype=abstract>
- Nussinov, R., Pieczenik, G., Griggs, J. R., & Kleitman, D. J. (1978). Algorithms for Loop Matchings. *SIAM Journal on Applied Mathematics*, 35(1), 68–82. <http://doi.org/10.1137/0135006>
- O'Neill, M., & Brabazon, A. (2006a). Grammatical Differential Evolution. *Proceedings of the 2006 International Conference on Artificial Intelligence, ICAI 2006, 1*, 231–236. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.3012.pdf>
- O'Neill, M., & Brabazon, A. (2006b). Grammatical Swarm: The generation of programs by social programming. *Natural Computing*, 5(4), 443–462. <http://doi.org/10.1007/s11047-006-9007-7>
- O'Neill, M., Leahy, F., & Brabazon, A. (2006). Grammatical swarm: A variable-length particle swarm algorithm. *Studies in Computational Intelligence*, 26, 59–74. http://doi.org/10.1007/978-3-540-33869-7_3
- O'Neill, M., McDermott, J., Swafford, J. M., Byrne, J., Hemberg, E., Brabazon, A., & Neill, M. O. (2010). Evolutionary Design using Grammatical Evolution and Shape Grammars: Designing a Shelter. *Int. J. Design Engineering*, 3(1), 4–24. <http://doi.org/doi:10.1504/IJDE.2010.032820>
- O'Neill, M., & Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4). <http://doi.org/10.1109/4235.942529>
- O'Neill, M., & Ryan, C. (2003). *Grammatical Evolution*. Boston, MA: Springer US. <http://doi.org/10.1007/978-1-4615-0447-4>
- O'Sullivan, John, C. R. (2002). An Investigation into the Use of Different Search Strategies with Grammatical Evolution. In J. A. Foster, E. Lutton, J. Miller, C. Ryan, & A. Tettamanzi (Eds.), *Genetic Programming, 5th European Conference, EuroGP 2002 Kinsale, Ireland, April 3–5, 2002 Proceedings* (Vol. 2278). Berlin, Heidelberg:

- Springer Berlin Heidelberg. <http://doi.org/10.1007/3-540-45984-7>
- Ortega, A., Dalhoum, A. A., & Alfonseca, M. (2003). Grammatical evolution to design fractal curves with a given dimension. *IBM Journal of Research and Development*, 47(4), 483–493. <http://doi.org/10.1147/rd.474.0483>
- Peignier, S., & Castañeta M., H. (n.d.). Búsqueda de estructuras secundarias óptimas y subóptimas de una cadena de ARN utilizando inteligencia artificial. *Revista Boliviana de Química*, 29(2), 147–157. Retrieved from http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S0250-54602012000200004&lng=es&nrm=iso&tlng=es
- Pörschke, D. (1974). Model calculations on the kinetics of oligonucleotide double helix coil transitions. Evidence for a fast chain sliding reaction. *Biophysical Chemistry*, 2(2), 83–96. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/4433687>
- Rivas, E., & Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285(5), 2053–68. <http://doi.org/10.1006/jmbi.1998.2436>
- Rothlauf, F. (2006). *Representations for genetic and evolutionary algorithms. Representations for Genetic and Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://doi.org/10.1007/3-540-32444-5>
- Ryan, C., Nicolau, M., & Neill, M. O. (2002). Genetic Algorithms Using Grammatical Evolution. In *Genetic Programming, 5th European Conference, EuroGP 2002 Kinsale, Ireland, April 3–5, 2002 Proceedings* (pp. 278–287). http://doi.org/10.1007/3-540-45984-7_27
- Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjolander, K., Underwood, R. C., & Haussler, D. (1994). Recent Methods for {RNA} Modeling Using Stochastic Context-Free Grammars. *Proc. 5th Symp. Combinatorial Pattern Matching*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.9031>
- Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C., & Haussler, D. (1994). Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Research*, 22(23), 5112–20. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=523785&tool=pmcentre&rendertype=abstract>
- Schrödinger, L. (2010). The PyMOL Molecular Graphics System, Version 1.7.4. Retrieved from <https://www.pymol.org/>
- Searls, D. B. (1992). The Linguistics of DNA. *American Scientist*, 80(6), pp. 579–591. Retrieved from <http://www.jstor.org/stable/29774782>
- Shaker, N. (2012). Evolving Personalized Content for Super Mario Bros Using Grammatical

Evolution. *Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE 12)*.

- Shaker, N., Nicolau, M., Yannakakis, G. N., Togelius, J., & O'Neill, M. (2012). Evolving levels for Super Mario Bros using grammatical evolution. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)* (pp. 304–311). IEEE. <http://doi.org/10.1109/CIG.2012.6374170>
- Shapiro, B. A., Bengali, D., Kasprzak, W., & Wu, J. C. (2001). RNA folding pathway functional intermediates: their prediction and analysis. *Journal of Molecular Biology*, *312*(1), 27–44. <http://doi.org/10.1006/jmbi.2001.4931>
- Shapiro, B. A., Wu, J. C., Bengali, D., & Potts, M. J. (2001). The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation. *Bioinformatics (Oxford, England)*, *17*(2), 137–48. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/11238069>
- Shapiro, J. (2001). Genetic Algorithms in Machine Learning. In C. Paliouras, Georgios and Karkaletsis, Vangelis and Spyropoulos (Ed.), *Machine Learning and Its Applications* (pp. 146–168). Springer Berlin Heidelberg. http://doi.org/10.1007/3-540-44673-7_7
- Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J., & Giegerich, R. (2006). RNASHAPES: an integrated RNA analysis package based on abstract shapes. *Bioinformatics (Oxford, England)*, *22*(4), 500–3. <http://doi.org/10.1093/bioinformatics/btk010>
- Turner, D. H., & Mathews, D. H. (2010). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, *38*(Database issue), D280–2. <http://doi.org/10.1093/nar/gkp892>
- Turner, D. H., Sugimoto, N., Jaeger, J. A., Longfellow, C. E., Freier, S. M., & Kierzek, R. (1987). Improved parameters for prediction of RNA structure. *Cold Spring Harbor Symposia on Quantitative Biology*, *52*, 123–33. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/2456874>
- van Batenburg, F. H., Gulyaev, A. P., & Pleij, C. W. (1995). An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *Journal of Theoretical Biology*, *174*(3), 269–80. <http://doi.org/10.1006/jtbi.1995.0098>
- Wang, G., Zhang, W., Ning, Q., & Chen, H. (2013). A Novel Framework Based on ACO and PSO for RNA Secondary Structure Prediction. *Mathematical Problems in Engineering*, *2013*, 1–8. <http://doi.org/10.1155/2013/796304>
- WATSON, J. D., & CRICK, F. H. C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, *171*(4356), 737–738. <http://doi.org/10.1038/171737a0>
- Wj Anderson, J., Tataru, P., Staines, J., Hein, J., & Lyngsø, R. (2012). Evolving stochastic

- context--free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 13, 78. <http://doi.org/10.1186/1471-2105-13-78>
- Xia, T., SantaLucia, J., Burkard, M. E., Kierzek, R., Schroeder, S. J., Jiao, X., ... Turner, D. H. (1998). Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37(42), 14719–35. <http://doi.org/10.1021/bi9809425>
- Yu, J., Zhang, C., Liu, Y., & Li, X. (2010). Simulating the Folding Pathway of RNA Secondary Structure Using the Modified Ant Colony Algorithm. *Journal of Bionic Engineering*, 7(4), 382–389. [http://doi.org/10.1016/S1672-6529\(10\)60270-3](http://doi.org/10.1016/S1672-6529(10)60270-3)
- Žegklitz, J., & Pošík, P. (2015). Symbolic Regression by Grammar-based Multi-Gene Genetic Programming. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference - GECCO Companion '15* (pp. 1217–1220). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2739482.2768484>
- Zuker, M. (1989). Computer prediction of RNA structure. *Methods in Enzymology*, 180, 262–88. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/2482418>
- Zuker, M. (2000). Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10(3), 303–310. [http://doi.org/10.1016/S0959-440X\(00\)00088-9](http://doi.org/10.1016/S0959-440X(00)00088-9)
- Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13), 3406–3415. <http://doi.org/10.1093/nar/gkg595>
- Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13), 3406–3415. Retrieved from <http://www.nar.oupjournals.org/cgi/doi/10.1093/nar/gkg595>
- Zuker, M., Jaeger, J. A., & Turner, D. H. (1991). A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Research*, 19(10), 2707–14. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=328190&tool=pmcentrez&rendertype=abstract>
- Zuker, M., & Sankoff, D. (1984). RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4), 591–621. <http://doi.org/10.1007/BF02459506>
- Zuker, M., & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1), 133–48. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=326673&tool=pmcentrez&rendertype=abstract>

Anexo A: Gramática para el modelamiento de estructuras secundarias del ARN

La primera gramática diseñada para el presente trabajo permite modelar las estructuras secundarias de moléculas de ARN, utilizando la representación en notación de horquillas. Las cadenas generadas por esta gramática incorporan información tanto de la secuencia de nucleótidos (estructura primaria), como de las estructuras secundarias conformadas gracias a las interacciones entre los elementos que hacen parte de la molécula.

Definición:

Sea $G_{rna} = \{\Sigma_{rna}, V_{rna}, Q_{rna}, e\}$ una gramática que genera cadenas que representan estructuras secundarias válidas bajo el modelo de representación en notación de horquillas. Donde Σ_{rna} es el conjunto de símbolos terminales (*alfabeto*) sobre el cual se generan las estructuras, V_{rna} es el conjunto de símbolos no terminales, Q_{rna} es el conjunto de reglas de producción y e constituye el símbolo inicial.

Sea Σ_b el alfabeto utilizado para representar de bases nitrogenadas que conforman una secuencia de ARN.

$$\Sigma_b = \{c, g, a, u\}$$

Sea Σ_p el alfabeto que representa las posibles combinaciones de pares de bases presentes en la estructura secundaria del ARN.

$$\Sigma_p = \{X, Y, Z, W, M, N\}$$

Sea Σ_R el conjunto de símbolos que delimitan las estructuras tipo horquilla (*Stem-loop*).

$$\Sigma^R = \{[, :,]\}$$

El conjunto de símbolos terminales de la gramática para generar cadenas que representen secundarias se define de la siguiente forma:

$$\Sigma_{rna} = B \cup P \cup R = \{c, g, a, u, X, Y, Z, W, M, N, [, :,]\}$$

El conjunto de símbolos no terminales V_{rna} se define como:

$$V_{rna} = \{d, h, s, l, p, q, b, l_e, l_i, l_m, l_h, l_b, l_s, l_{rb}, l_{sb}, i, j, k, m, n, r, o\}$$

Se define un conjunto de meta reglas de producción Q_{rna} de la forma $\alpha \rightarrow \beta$, donde α y β son cadenas de símbolos en $\Sigma_{rna} \cup V_{rna}$. Las reglas son aplicadas de manera progresiva para generar estructuras secundarias válidas en el contexto de la representación en notación de horquillas.

Las reglas se muestran a continuación agrupadas según el tipo de motivo estructural que modelan:

Estructura secundaria

$$e \rightarrow d \mid h$$

Tallo (*Stack*)

$$s \rightarrow pq$$

$$q \rightarrow pq \mid p$$

Bases libres en los extremos 5' o 3' (*Dangling Ends*)

$$d \rightarrow [:l]$$

Pares de bases (*Base pairs*)

$$p \rightarrow X \mid Y \mid Z \mid W \mid M \mid N$$

Horquilla (*Hairpin*)

$$h \rightarrow [s:l]$$

Bucles (*Loops*)

$$l \rightarrow l_e \mid l_i \mid l_m \mid l_h$$

Bucle Externo (*External Loop*)

$$l_e \rightarrow k \mid kr$$

$$r \rightarrow m \mid mr \mid \varepsilon$$

Bucles horquilla (*Hairpin Loops*)

$$l_h \rightarrow bi$$

$$i \rightarrow bj$$

$$j \rightarrow bk \mid b$$

Bucles internos (*Internal Loops*)

$$l_i \rightarrow l_b \mid l_s$$

$$l_b \rightarrow l_{rb} \mid l_{lb}$$

$$l_s \rightarrow jl_{rb} \mid l_{lb}j$$

$$l_{rb} \rightarrow hj$$

$$l_{lb} \rightarrow jh$$

Bases libres (*Single-stranded regions*)

$$k \rightarrow bk \mid \varepsilon$$

$$b \rightarrow c \mid g \mid a \mid u$$

Bucles multi-rama (*Multiloops*)

$$l_m \rightarrow ko$$

$$o \rightarrow mn$$

$$m \rightarrow h \mid hk$$

$$n \rightarrow m \mid mn$$

Sea $L(G_{rna})$ el lenguaje generado por G_{rna} definido de la siguiente forma:

$$L(G_{rna}) = \{w \mid w \in \Sigma_{rna}^*, s \Rightarrow^* w\}$$

El lenguaje $L(G_{rna})$ es el conjunto de todas las estructuras secundarias de RNA válidas representadas en notación de horquillas.

Anexo B: Ejemplo de la generación de estructuras secundarias de ARN utilizando la gramática propuesta.

Este anexo muestra un ejemplo de la derivación de una estructura secundaria de ARN utilizando las reglas de producción de la gramática definida en el Anexo A.

Sea la siguiente estructura secundaria representada en notación de puntos notación de puntos y paréntesis (*Dot-Bracket Notation*).

UGCCGAGAGGUCACCUCCGCCACCAUUGGCUCGCAUGUUG
 (((.((((((...))))).((((.....))))))))).....

La representación de la estructura secundaria en notación de horquillas (*stem-loops*) se define a continuación:

[: [WYX : c [YZ : [ZY Y : uca] c [YXXZ : ccau]]] uguug]

La estructura secundaria puede ser derivada a partir de la aplicación de las reglas de producción de la gramática G_{rna} definida en el Anexo A.

$e \rightarrow d$
 $d \rightarrow [: l]$
 $[: l] \rightarrow [: l_e]$
 $[: l_e] \rightarrow [: kr]$
 $[: kr] \rightarrow [: khk]$
 $[: khk] \rightarrow [: khbk]$
 $[: khbk] \rightarrow [: khbbk]$
 $[: khbbk] \rightarrow [: khbbb k]$
 $[: khbbb k] \rightarrow [: khbbbb k]$
 $[: khbbbb k] \rightarrow [: khbbbbb k]$
 $[: khbbbbb k] \rightarrow [: khbbbbb \varepsilon]$
 $[: khbbbbb \varepsilon] \rightarrow [: khbbbbb]$
 $[: khbbbbb] \rightarrow [: khbbbbb g]$
 $[: khbbbbb g] \rightarrow [: khbbbbb u]$
 $[: khbbbbb u] \rightarrow [: khbbuuu]$
 $[: khbbuuu] \rightarrow [: khbbuuu g]$
 $[: khbbuuu g] \rightarrow [: khbbuuu u]$
 $[: khbbuuu u] \rightarrow [: khbbuuu u g]$
 $[: khbbuuu u g] \rightarrow [: k[s : l]uuuu]$
 $[: k[s : l]uuuu] \rightarrow [: k[s : l_i]uuuu]$
 $[: k[s : l_i]uuuu] \rightarrow [: k[s : l_b]uuuu]$
 $[: k[s : l_i]uuuu] \rightarrow [: k[s : l_b]uuuu]$
 $[: k[s : l_b]uuuu] \rightarrow [: k[s : l_{lb}]uuuu]$
 $[: k[s : l_{lb}]uuuu] \rightarrow [: k[s : jh]uuuu]$
 $[: k[s : jh]uuuu] \rightarrow [: k[s : j[s : l]]uuuu]$
 $[: k[s : j[s : l]]uuuu] \rightarrow [: k[s : j[s : l_m]]uuuu]$
 $[: k[s : j[s : l_m]]uuuu] \rightarrow [: k[s : j[s : ko]]uuuu]$
 $[: k[s : j[s : ko]]uuuu] \rightarrow [: k[s : j[s : kmn]]uuuu]$
 $[: k[s : j[s : kmn]]uuuu] \rightarrow [: k[s : j[s : kmm]]uuuu]$
 $[: k[s : j[s : kmm]]uuuu] \rightarrow [: k[s : j[s : kmh]]uuuu]$
 $[: k[s : j[s : kmh]]uuuu] \rightarrow [: k[s : j[s : km[s : l]]]uuuu]$
 $[: k[s : j[s : km[s : l]]]uuuu] \rightarrow [: k[s : j[s : km[s : l_h]]]uuuu]$
 $[: k[s : j[s : km[s : l_h]]]uuuu] \rightarrow [: k[s : j[s : km[s : bi]]]uuuu]$
 $[: k[s : j[s : km[s : bi]]]uuuu] \rightarrow [: k[s : j[s : km[s : bbj]]]uuuu]$
 $[: k[s : j[s : km[s : bbj]]]uuuu] \rightarrow [: k[s : j[s : km[s : bbbk]]]uuuu]$

$[:k[s:j[s:km[s:bbbk]]]uguug] \rightarrow [:k[s:j[s:km[s:bbbbk]]]uguug]$
 $[:k[s:j[s:km[s:bbbbk]]]uguug] \rightarrow [:k[s:j[s:km[s:bbbbe]]]uguug]$
 $[:k[s:j[s:km[s:bbbbe]]]uguug] \rightarrow [:k[s:j[s:km[s:bbbbb]]]uguug]$
 $[:k[s:j[s:km[s:bbbbb]]]uguug] \rightarrow [:k[s:j[s:km[s:bbbu]]]uguug]$
 $[:k[s:j[s:km[s:bbbu]]]uguug] \rightarrow [:k[s:j[s:km[s:bbau]]]uguug]$
 $[:k[s:j[s:km[s:bbau]]]uguug] \rightarrow [:k[s:j[s:km[s:bcau]]]uguug]$
 $[:k[s:j[s:km[s:bcau]]]uguug] \rightarrow [:k[s:j[s:km[s:ccau]]]uguug]$
 $[:k[s:j[s:km[s:ccau]]]uguug] \rightarrow [:k[s:j[s:km[pq:ccau]]]uguug]$
 $[:k[s:j[s:km[pq:ccau]]]uguug] \rightarrow [:k[s:j[s:km[ppq:ccau]]]uguug]$
 $[:k[s:j[s:km[ppq:ccau]]]uguug] \rightarrow [:k[s:j[s:km[pppq:ccau]]]uguug]$
 $[:k[s:j[s:km[pppq:ccau]]]uguug] \rightarrow [:k[s:j[s:km[pppp:ccau]]]uguug]$
 $[:k[s:j[s:km[pppp:ccau]]]uguug] \rightarrow [:k[s:j[s:km[pppz:ccau]]]uguug]$
 $[:k[s:j[s:km[pppz:ccau]]]uguug] \rightarrow [:k[s:j[s:km[ppxz:ccau]]]uguug]$
 $[:k[s:j[s:km[ppxz:ccau]]]uguug] \rightarrow [:k[s:j[s:km[pxxz:ccau]]]uguug]$
 $[:k[s:j[s:km[pxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:km[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:km[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khk[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khk[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khhk[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khhk[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khhb[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khhb[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khhbe[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khhbe[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khh[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khh[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:khc[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:khc[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:l]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:l]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:l_h]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:l_h]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:bi]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:bi]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:bbj]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:bbj]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:bbb]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:bbb]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:bba]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:bba]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:bca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:bca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[s:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[s:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[pq:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[pq:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[ppq:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[ppq:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[pppq:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[pppq:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[pppp:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[pppp:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[pppy:uca]c[yxxz:ccau]]]uguug]$
 $[:k[s:j[s:k[pppy:uca]c[yxxz:ccau]]]uguug] \rightarrow [:k[s:j[s:k[ppyy:uca]c[yxxz:ccau]]]uguug]$

$[:k[s:j[s:k[ppYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[s:k[pZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[s:k[pZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[s:k[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[s:k[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[s:\varepsilon[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[s:\varepsilon[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[s:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[s:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[pq:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[pq:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[pp:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[pp:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[pz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[pz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:j[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:j[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:b[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:b[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[s:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[s:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[pq:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[pq:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[ppq:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[ppq:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[ppp:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[ppp:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[ppx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[ppx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[pyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[pyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[wyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:k[wyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:k[wyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$
 $[:\varepsilon[wyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug] \rightarrow [:wyx:c[yz:[YZYY:uca]c[YXXX:ccau]]]uguug]$

Anexo C: Gramática para el modelamiento del plegamiento de secuencias de ARN

La segunda gramática propuesta, modela operaciones elementales que conforman un conjunto de movimientos (*move set*) aplicables a estructuras representadas en notación de horquillas. Esto con el fin de simular la formación y destrucción de pares de bases que acontecen lo largo del trayecto de plegamiento (*folding path*) que recorre la molécula durante el proceso de conformación de las estructuras secundarias finales.

Definición:

Sea $G_{fold} = \{\Sigma_{rna}, V_{fold}, Q_{fold}, F_{fold}, e\}$ una gramática que transforma cadenas que representan estructuras secundarias válidas bajo el modelo de representación en notación de horquillas generadas a partir de la gramática G_{rna} . Donde Σ_{rna} es el alfabeto sobre el cual se generan las estructuras en notación de horquillas (definido junto con la gramática G_{rna}), V_{fold} es el conjunto de símbolos no terminales, Q_{fold} es el conjunto de reglas de producción que modelan la formación y disrupción de pares de bases, F_{fold} es un conjunto de funciones que transforma elementos del conjunto de símbolos terminales Σ_{rna} en elementos del conjunto V_{fold} de elementos no terminales, y viceversa. Y e constituye el símbolo inicial, que para el caso de esta gramática será cualquier cadena perteneciente al lenguaje de estructuras secundarias $L(G_{rna})$.

El conjunto de símbolos no terminales de la gramática V_{fold} , se define a partir de la aplicación de los operadores \rightarrow y \leftarrow al alfabeto Σ_b ; y el operador \sim al alfabeto Σ_p . Los operadores \rightarrow y \leftarrow se definen como marcadores para nucleótidos complementarios

candidatos a conformar pares de bases dentro de una estructura, y el operador \sim se define como un marcador para los pares de bases ya conformados que son candidatos a suprimirse.

$$V_{fold} = \overrightarrow{\Sigma_b} \cup \overleftarrow{\Sigma_b} \cup \widetilde{\Sigma_p} = \{\vec{c}, \vec{g}, \vec{a}, \vec{u}, \vec{c}, \vec{g}, \vec{a}, \vec{u}, \vec{X}, \vec{Y}, \vec{Z}, \vec{W}, \vec{M}, \vec{N}\}$$

F_{fold} está conformado por un conjunto de funciones que permite obtener la equivalencia entre un par de bases y los nucleótidos que lo conforman. Para la gramática se definieron un total de tres funciones para modelar las combinaciones de nucleótidos que pueden conformar un par de bases.

Sean $\gamma, \mu \in \Sigma_b$, se define $\Psi(\gamma, \mu)$ como una función de emparejamiento de nucleótidos según su posición:

$$\Psi(\gamma, \mu) = \begin{cases} X & \text{if } \gamma = c \wedge \mu = g \\ Y & \text{if } \gamma = g \wedge \mu = c \\ Z & \text{if } \gamma = a \wedge \mu = u \\ W & \text{if } \gamma = u \wedge \mu = a \\ M & \text{if } \gamma = g \wedge \mu = u \\ N & \text{if } \gamma = u \wedge \mu = g \end{cases}$$

Sea $\alpha \in \Sigma_p$, se definen las funciones $\Phi_5(\alpha)$ y $\Phi_3(\alpha)$ como funciones para obtener los nucleótidos en la dirección 5' y 3' que conforman el par de bases α .

$$\Phi_5(\alpha) = \begin{cases} c & \text{if } \alpha = X \\ g & \text{if } \alpha = Y \vee i = M \\ a & \text{if } \alpha = Z \\ u & \text{if } \alpha = W \vee i = N \end{cases} \quad \Phi_3(\alpha) = \begin{cases} c & \text{if } \alpha = Y \\ g & \text{if } \alpha = X \vee i = N \\ a & \text{if } \alpha = W \\ u & \text{if } \alpha = Z \vee i = M \end{cases}$$

Todo proceso de conformación de estructuras secundarias de ARN se puede definir como una secuencia de estados dentro del espacio de estructuras posibles que puede adoptar la secuencia. El tránsito de un estado a otro se determina por la aplicación de operaciones que transforman las estructuras generando una especie de trayectoria de plegamiento (*folding path*) el cual inicia a partir de un estado sin estructuras hasta el estado en el cual la molécula adquiere su plegamiento final. En este contexto, toda trayectoria de plegamiento está

conformada por una serie de inserciones y eliminaciones de pares de bases. El conjunto de reglas de producción Q_{fold} modela este conjunto de operaciones elementales (*Move Set*) para simular la trayectoria de plegamiento a medida que se ejecuta el proceso de derivación.

i. Creación de un par de bases

Sea L un bucle dentro de una estructura secundaria, y dos nucleótidos γ y $\mu \in \Sigma_b$ en los extremos 5' y 3' del bucle respectivamente:



Los elementos γ y μ son transformados de símbolos terminales a símbolos no terminales al aplicar sobre ellos los operadores $\bar{}$ y $\tilde{}$ respectivamente:



Reglas de producción:

$$\bar{\gamma} L \tilde{\mu} \rightarrow [\Psi(\gamma, \mu): L] \Leftrightarrow \gamma \wedge \mu \text{ son bases complementarias}$$

$$\bar{\gamma} L \tilde{\mu} \rightarrow \gamma L \mu \Leftrightarrow \gamma \wedge \mu \text{ son bases complementarias}$$

$$:\bar{\gamma} L \tilde{\mu}] \rightarrow \Psi(\gamma, \mu): L] \Leftrightarrow \gamma \wedge \mu \text{ son bases complementarias}$$

$$:\bar{\gamma} L \tilde{\mu}] \rightarrow : \gamma L \mu] \Leftrightarrow \gamma \wedge \mu \text{ no son bases complementarias}$$

$$:\bar{\gamma}[\alpha^n: L]\tilde{\mu}] \rightarrow \Psi(\gamma, \mu)\alpha^n: L]; \alpha \in \Sigma_p; n \geq 2 \Leftrightarrow \gamma \wedge \mu \text{ son bases complementarias}$$

$$:\bar{\gamma}[\alpha^n: L]\tilde{\mu}] \rightarrow : \gamma[\alpha^n: L]\mu]; \alpha \in \Sigma_p; n \geq 2 \Leftrightarrow \gamma \wedge \mu \text{ no son bases complementarias}$$

ii. Ruptura de un par de bases

Dada una estructura de tipo horquilla $[\alpha^m : L] = [\alpha_1 \alpha_2 \dots \alpha_j \dots \alpha_{m-1} \alpha_m : L]$; $\alpha_{1, \dots, m} \in \Sigma_p$. Un par de bases α_j es transformado de un elemento terminal a un símbolo no terminal al aplicar sobre él el operador \sim .

Reglas de producción:

$$[\alpha_1 \alpha_2 \dots \tilde{\alpha}_j \dots \alpha_{m-1} \alpha_m : L] \rightarrow [\alpha_1 \alpha_2 \dots \alpha_{j-1} : \Phi_5(\alpha_j) [\alpha_{j+1} \dots \alpha_{m-1} \alpha_m : L] \Phi_3(\alpha_j)]$$

$$\begin{aligned}
& [\alpha_1 \alpha_2 \dots \tilde{\alpha}_m : L] \rightarrow [\alpha_1 \alpha_2 \dots \alpha_{m-1} : \Phi_5(\alpha_m) L \Phi_3(\alpha_m)] \\
& \gamma[\tilde{\alpha}_1 \alpha_2 \dots \alpha_m : L] \mu \rightarrow \gamma \Phi_5(\alpha_1) [\alpha_2 \dots \alpha_m : L] \Phi_3(\alpha_1) \mu ; \gamma, \mu \in \Sigma_b \\
& [\tilde{\alpha}_1 \alpha_2 \dots \alpha_m : L] \rightarrow [: \Phi_5(\alpha_1) [\alpha_2 \dots \alpha_m : L] \Phi_3(\alpha_1)] \\
& [\alpha : \rightarrow [\tilde{\alpha} : \\
& \gamma[\tilde{\alpha} : L] \mu \rightarrow \gamma \Phi_5(\alpha) L \Phi_3(\alpha) \mu ; \gamma, \mu \in \Sigma_b \\
& [\tilde{\alpha} : L] \rightarrow [: \Phi_5(\alpha) L \Phi_3(\alpha)]
\end{aligned}$$

La gramática también contempla reglas de producción para operaciones más complejas como la formación y disrupción de tallos. La formación de hélices a partir de regiones de nucleótidos complementarios, se modela a partir de un concepto conocido como mecanismo de cierre (*zipper mechanism*) (Pörschke, 1974), el cual plantea una analogía entre el apilamiento de pares de bases (o ruptura de los mismos), y el proceso de apertura o cierre de una cremallera.

iii. Construcción de un tallo

Sea L un bucle dentro de una estructura secundaria, y γ^n y $\mu^m \in \Sigma_b$ con $n, m \geq 2$, dos regiones de nucleótidos libres en los extremos 5' y 3' del bucle respectivamente:

$$\gamma^n L \mu^m$$

$$\gamma_1 \gamma_2, \dots, \gamma_{n-1} \gamma_n L \mu_1 \mu_2, \dots, \mu_{m-1} \mu_m$$

Reglas de producción:

Cierre hacia arriba (*Zip up*)

$$\vec{\gamma}_1 \gamma_2, \dots, \gamma_{n-1} \gamma_n L \mu_1 \mu_2, \dots, \mu_{m-1} \vec{\mu}_m$$

$$\begin{aligned}
& \vec{\gamma}_1, \dots, \gamma_n L \mu_1, \dots, \vec{\mu}_m \rightarrow [\Psi(\vec{\gamma}_1, \vec{\mu}_m) : \vec{\gamma}_2, \dots, \gamma_n L \mu_1, \dots, \vec{\mu}_{m-1}] \\
& \Leftrightarrow \gamma_1 \gamma_2 \wedge \mu_{m-1} \mu_m \text{ son regiones complementarias}
\end{aligned}$$

Cierre hacia abajo (*Zip down*)

$$\gamma_1, \dots, \gamma_{n-1} \vec{\gamma}_n L \vec{\mu}_1 \mu_2, \dots, \mu_m$$

$$\begin{aligned} \gamma_1, \dots, \vec{\gamma}_n L \vec{\mu}_1, \dots, \mu_m &\rightarrow \gamma_1, \dots, \vec{\gamma}_{n-1} [\Psi(\vec{\gamma}_n, \vec{\mu}_1): L] \vec{\mu}_2, \dots, \mu_m \\ \Leftrightarrow \gamma_{n-1} \gamma_n \wedge \mu_1 \mu_2 &\text{ son regiones complementarias} \end{aligned}$$

Cierre en ambas direcciones (*Zip up and down*)

Dado un bucle L dentro de una estructura, y dos regiones de nucleótidos libres γ^n y $\mu^m \in \Sigma_b$ con $n, m \geq 3$ en los extremos 5' y 3' del bucle respectivamente:

$$\gamma_1, \dots, \gamma_{i-1} \gamma_i \gamma_{i+1}, \dots, \gamma_n L \mu_1, \dots, \mu_{j-1} \mu_j \mu_{j+1}, \dots, \mu_m$$

$$\gamma_1, \dots, \gamma_{i-1} \vec{\gamma}_i \gamma_{i+1}, \dots, \gamma_n L \mu_1, \dots, \mu_{j-1} \vec{\mu}_j \mu_{j+1}, \dots, \mu_m$$

$$\begin{aligned} \gamma_1, \dots, \gamma_{i-1} \vec{\gamma}_i \gamma_{i+1}, \dots, \gamma_n L \mu_1, \dots, \mu_{j-1} \vec{\mu}_j \mu_{j+1}, \dots, \mu_m \\ \rightarrow \gamma_1, \dots, \vec{\gamma}_{i-1} [\Psi(\gamma_i, \mu_j): \vec{\gamma}_{i+1}, \dots, \gamma_n L \mu_1, \dots, \vec{\mu}_{j-1}] \vec{\mu}_{j+1}, \dots, \mu_m \\ \Leftrightarrow \gamma_{i-1} \gamma_i \gamma_{i+1} \wedge \mu_{j-1} \mu_j \mu_{j+1} &\text{ son regiones complementarias} \end{aligned}$$

iv. Unión de tallos

Sean dos estructuras tipo horquillas anidadas de la forma:

$$[: [S : L]]$$

ó

$$[\alpha^n : [\beta^m : L]]; \alpha_{1,\dots,n}, \beta_{1,\dots,m} \in \Sigma_p ; n, m \geq 2$$

Reglas de producción:

$$[: [S : L]] \rightarrow [S : L]$$

$$[\alpha^n : [\beta^m : L]] \rightarrow [\alpha^n \beta^m : L] ; \alpha_{1,\dots,n}, \beta_{1,\dots,m} \in \Sigma_p ; n, m \geq 2$$

v. Disrupción de tallos

Dada una estructura tipo horquilla:

$$[\alpha^m : L] = [\alpha_1 \alpha_2 \dots \alpha_j \dots \alpha_{m-1} \alpha_m : L] ; \alpha_{1,\dots,m} \in \Sigma_b ; m \geq 2$$

Reglas de producción:

Cierre hacia arriba (*Zip up*)

$$[\tilde{\alpha}_1 \alpha_2 \dots \alpha_{m-1} \alpha_m : L] \rightarrow \Phi_5(\alpha_1) [\tilde{\alpha}_2 \dots \alpha_{m-1} \alpha_m : L] \Phi_3(\alpha_1)$$

Cierre hacia abajo (*Zip down*)

$$[\alpha_1 \alpha_2 \dots \alpha_{m-1} \tilde{\alpha}_m : L] \rightarrow [\alpha_1 \alpha_2 \dots \tilde{\alpha}_{m-1} : \Phi_5(\alpha_m) L \Phi_3(\alpha_m)]$$

Anexo D: Derivaciones del conjunto de operaciones (*Move set*) definidas para modelar el plegamiento de secuencias de ARN

En el presente anexo se presentan ejemplos de las derivaciones del conjunto de operaciones definidas sobre estructuras secundarias de ARN, aplicando las reglas de producción de la gramática G_{fold} para modelar la trayectoria de plegamiento de la molécula.

Operaciones elementales

i. Construcción de un par de bases

UGCCGAGAGGUCACCUCCGCCACCAUUGGCUCGCAUGUUG
 (((.(((.....)))..(((.....)).))).....
 ↓
 (((.(((.....)))..(((.....)).))).....

Derivación:

$$\begin{aligned}
 [: [WYX : c [YZ : [YZYY : uca] cg [XXZ : cc au] c]] uguug] &\rightarrow [: [WYX : c [YZ : [YZYY : uca] c\vec{g} [XXZ : cc au] \vec{c}]] uguug] \\
 [: [WYX : c [YZ : [YZYY : uca] c\vec{g} [XXZ : cc au] \vec{c}]] uguug] &\rightarrow [: [WYX : c [YZ : [YZYY : uca] c [\Psi(g, c)XXZ : cc au]]] uguug] \\
 [: [WYX : c [YZ : [YZYY : uca] c [\Psi(g, c)XXZ : cc au]]] uguug] &\rightarrow [: [WYX : c [YZ : [YZYY : uca] c [YXXX : cc au]]] uguug]
 \end{aligned}$$

ii. Ruptura de un par de bases

UGCCGAGAGGUCACCUCCGCCACCAUUGGCUCGCAUGUUG
 (((.(((.....)))..(((.....)).))).....
 ↓
 (((.(((.....)))..(((.....)).))).....

Derivación:

$$\begin{aligned}
 [: [WYX : c [YZ : [YZYY : uca] cg [XXZ : cc au] c]] uguug] &\rightarrow [: [WYX : c [YZ : [\tilde{Y}ZYY : uca] cg [XXZ : cc au] c]] uguug] \\
 [: [WYX : c [YZ : [\tilde{Y}ZYY : uca] cg [XXZ : cc au] c]] uguug] &\rightarrow [: [WYX : c [YZ : \Phi_5(Y) [ZYY : uca] \Phi_3(Y) cg [XXZ : cc au] c]] uguug] \\
 [: [WYX : c [YZ : \Phi_5(Y) [YZYY : uca] \Phi_3(Y) cg [XXZ : cc au] c]] uguug] &\rightarrow [: [WYX : c [YM : g [ZYY : uca] c cg [XXZ : cc au] c]] uguug]
 \end{aligned}$$

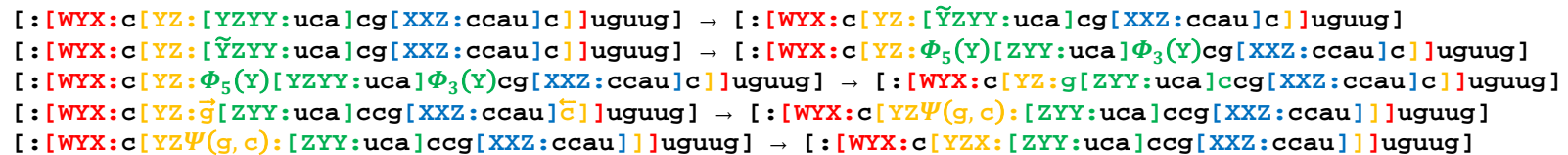
iii. Cambio de un par de bases

Esta operación se define como la ruptura de un par de bases seguido por la conformación de uno nuevo, en el cual se encuentra involucrado uno de los nucleótidos del par de bases original. La operación de ruptura puede presentarse en el primer o último par de bases del tallo de una horquilla.

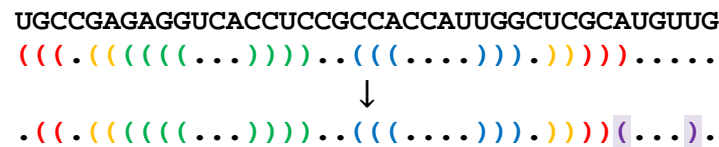
a. Caso 1: Cuando la ruptura se presenta en el último par de bases del tallo



Derivación:



b. Caso 2: Cuando la ruptura se presenta en el primer par de bases del tallo

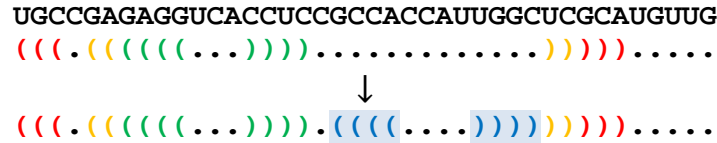


Derivación:

$$\begin{aligned}
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: [\tilde{\text{WYX}} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \\
 & [: [\tilde{\text{WYX}} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{uguug}] \rightarrow [: \Phi_5(\text{w}) [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \Phi_3(\text{w}) \text{uguug}] \\
 & [: \Phi_5(\text{w}) [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \Phi_3(\text{w}) \text{uguug}] \rightarrow [: \text{u} [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \text{a} \text{uguug}] \\
 & [: \text{u} [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] \tilde{\text{a}} \text{uguug}] \rightarrow [: \text{u} [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] [\Psi(\text{a}, \text{u}) : \text{UGU}] \text{g}] \\
 & [: \text{u} [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] [\Psi(\text{a}, \text{u}) : \text{UGU}] \text{g}] \rightarrow [: \text{u} [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} [\text{XXZ} : \text{ccau}] \text{c}]] [\text{Z} : \text{UGU}] \text{g}]
 \end{aligned}$$

Operaciones complejas

iv. Construcción de un tallo



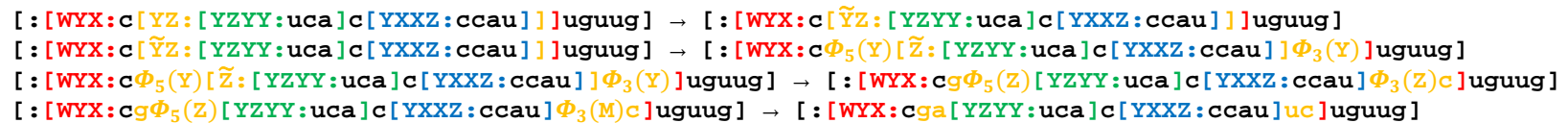
Derivación:

$$\begin{aligned}
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgccacc} \text{cau} \text{uggc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgcc} \tilde{\text{a}} \text{ccau} \tilde{\text{u}} \text{ggc}]] \text{uguug}] \\
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cgcc} \tilde{\text{a}} \text{ccau} \tilde{\text{u}} \text{ggc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \tilde{\text{c}} [\Psi(\text{a}, \text{u}) : \text{ccau}] \tilde{\text{g}} \text{gc}]] \text{uguug}] \\
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \tilde{\text{c}} [\Psi(\text{a}, \text{u}) : \text{ccau}] \tilde{\text{g}} \text{gc}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \tilde{\text{c}} [\Psi(\text{c}, \text{g}) \text{z} : \text{ccau}] \tilde{\text{g}} \text{c}]] \text{uguug}] \\
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{cg} \tilde{\text{c}} [\Psi(\text{c}, \text{g}) \text{z} : \text{ccau}] \tilde{\text{g}} \text{c}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} \tilde{\text{g}} [\Psi(\text{c}, \text{g}) \text{xz} : \text{ccau}] \tilde{\text{c}}]] \text{uguug}] \\
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} \tilde{\text{g}} [\Psi(\text{c}, \text{g}) \text{xz} : \text{ccau}] \tilde{\text{c}}]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\Psi(\text{g}, \text{c}) \text{xxz} : \text{ccau}]]] \text{uguug}] \\
 & [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\Psi(\text{g}, \text{c}) \text{xxz} : \text{ccau}]]] \text{uguug}] \rightarrow [: [\text{WYX} : \text{c} [\text{YZ} : [\text{YZYY} : \text{uca}] \text{c} [\text{YXXZ} : \text{ccau}]]] \text{uguug}]
 \end{aligned}$$

v. Disrupción de tallos



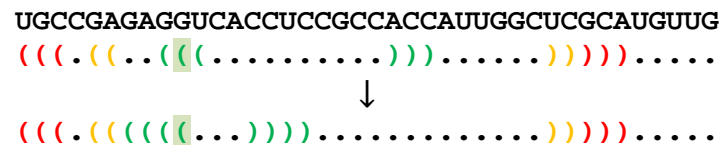
Derivación:



vi. Deslizamiento de tallos:

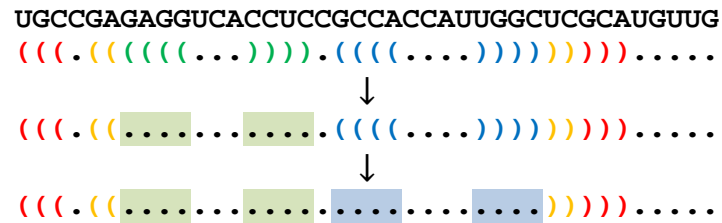
Este movimiento es análogo al movimiento de cambio de pares de bases. Se define como la destrucción de un tallo seguido por la formación de uno nuevo dentro del bucle generado luego de la destrucción del tallo original. Al menos uno de las bases que hacían parte del tallo eliminado deberá hacer parte del nuevo tallo. Existen también dos casos de este movimiento; el primero cuando el tallo se desliza hacia el bucle interno de la horquilla y el segundo cuando el deslizamiento se realiza hacia el bucle externo a la estructura tipo horquilla.

a. Caso 1: Deslizamiento hacia el bucle interno



vii. Disrupción de tallos en un mismo nivel:

Esta operación fuerza la eliminación de tallos que se encuentren en el mismo nivel (adyacentes al mismo bucle). Se implementa debido a que en ocasiones desaparición de múltiples tallos va seguido de la formación de un tallo más estable a partir de las bases que conformaban los tallos iniciales.



Anexo E: Muestras de las mejores trayectorias de plegamiento encontradas utilizando el algoritmo propuesto

A continuación se presentan muestras de las mejores trayectorias de plegamiento encontradas al utilizar el algoritmo propuesto utilizando los parámetros por defecto.

i. Resultados de las pruebas realizadas utilizando secuencias correspondientes a Riboswitches

Tabla 1. Mejor plegamiento obtenido para la secuencia de prueba identificada como rb1.

Etapa	Estructura Secundaria	ΔG
0	[: cacucauauaaucgcguggauauggcacgcaaguuuacuacggggcaccguaaauguccgacuaugggugagcaauggaaccgacguguacg guuuuuugugauaucagcauugcuugcucuuuuuuugagcgggcaaugcuuuuuuu]	0.00
1	[: [XZXNXZWZ : uaaucgcguggauauggcacgcaaguuuacuacggggcaccguaaauguccgac] agcaauggaaccgacguguacgguuuu uuugugauaucagcauugcuugcucuuuuuuugagcgggcaaugcuuuuuuu]	-5.83
2	[: [XZXNXZWZ : uaaucgcguggauauggcacgcaaguuuacuac [XYMXZ : ccguaaa] ac] agcaauggaaccgacguguacgguuuuuuug ugauaucagcauugcuugcucuuuuuuugagcgggcaaugcuuuuuuu]	-12.04
3	[: [XZXNXZWZ : uaaucgcguggauau [MMX : acgcaa] ucuaac [XYMXZ : ccguaaa] ac] agcaauggaaccgacguguacgguuuuuuug ugauaucagcauugcuugcucuuuuuuugagcgggcaaugcuuuuuuu]	-15.40
4	[: [XZXNXZWZ : uaaucgcguggauau [MMX : acgcaa] ucuaac [XYMXZ : ccguaaa] ac] agcaauggaaccgacguguacgguuuuuuug ugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuu]	-38.40
5	[: [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuacuac [XYMXZ : ccguaaa] ac] agcaauggaaccgacguguacgguuuuuuugug auauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuu]	-47.10
6	[: [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuacuac [XYMXZ : ccguaaa] ac] agcaauggaaccgacgug [NZXMM : uuuu] au auc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuu]	-48.20
7	[: [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuacuac [XYMXZ : ccguaaa] ac] agcaau [MMZXXY : cacgugua] uuugauau c [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuu]	-53.00
9	[: [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuacuac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZXXY : cacgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuu]	-55.70

Tabla 2. Mejor plegamiento obtenido para la secuencia de prueba identificada como rb2.

Etapa	Estructura Secundaria	ΔG
0	[:ggcuucauuaaauccuaauggauaugguuugggaguuucuaccaagagccguaaacucuugauuaugaagucugucgcuuuaucgaaauuuuauaaagagaagacucaugaau]	0.00
1	[: [YYXWN :cauuaaauccuaauggauaugguuugggaguuucuaccaa]guaaacucuugauuaugaagucugucgcuuuaucgaaauuuuauaaagagaagacucaugaau]	-4.18
2	[: [YMXWWXZWZ :uaauccuaauggauaugguuugggaguuucuaccaagagccguaaacucuugau]ugucgcuuuaucgaaauuuuauaaagagaagacucaugaau]	-8.23
3	[: [YMXWWXZWZ :uaa [WXXNZZ :ugauauggu]guuucuaccaagagccguaaacucuugau]ugucgcuuuaucgaaauuuuauaaagagaagacucaugaau]	-13.17
4	[: [YMXWWXZWZ :uaa [WXXNZZ :ugauauggu]guuucuac [XZZYZY :ccguaaa]au]ugucgcuuuaucgaaauuuuauaaagagaagacucaugaau]	-22.00
5	[: [YMXWWXZWZ :uaa [WXXNZZ :ugauauggu]guuucuac [XZZYZY :ccguaaa]au]ugucg [XWWWZW :ccgaaauuuu]agaagacucugaau]	-23.10
6	[: [YMXWWXZWZ :uaa [WXXNZZ :ugauauggu]guuucuac [XZZYZY :ccguaaa]au]ug [WX :g [XWWWZW :ccgaaauuuu]a]agacucaugaau]	-24.80

Tabla 3. Mejor plegamiento obtenido para la secuencia de prueba identificada como rb3.

Etapa	Estructura Secundaria	ΔG
0	[: cucuuaucaagagagguggagggacuggcccgauaaccggcaaccagccuuagggcauggugccaauccugcagcgguuucgcguugaaagaugagagauucuuuguagucucuucuuuuagcgaagggacuuuuuuu]	0.00
1	[: cucuuaucaagagagguggagggac [WYYX : ccgaugaaaccggcaaccagccuuagggcauggu] auuccugcagcgguuucgcguugaaagau gagagauucuuuguagucucuucuuuuagcgaagggacuuuuuuu]	-2.23
2	[: cucuuaucaagagagguggagggac [WYYX : ccgaugaaaccggca [ZXXZ : gccuuagggca]] auuccugcagcgguuucgcguugaaagaug agagauucuuuguagucucuucuuuuagcgaagggacuuuuuuu]	-5.40
3	[: cucuuaucaagagagguggagggac [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagaug agagauucuuuguagucucuucuuuuagcgaagggacuuuuuuu]	-12.10
4	[: cucuuaucaagagagguggagggac [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagaug agagauucuuuguagucucuucuuuuagcgaagggacuuuuuuu]	-15.20
5	[: cucuuaucaagagagguggagggac [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagau [YZYZ : gauucuuuguag] uucuuuuagcgaagggacuuuuuuu]	-16.90
6	[: cucuuaucaag [ZMZY : uggagggac [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaaga u [YZYZ : gauucuuuguag] uu] agcgaagggacuuuuuuu]	-18.20
7	[: cucuuaucaagagagg [WMMZMYMZ : c [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaaga u [YZYZ : gauucuuuguag]] gcaagggacuuuuuuu]	-19.80
8	[: [XWXWN : aucaa] g [WMMZMYMZ : c [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagau [YZYZ : gauucuuuguag]] gcaagggacuuuuuuu]	-23.70
9	[: [XWXWN : aucaa] g [WMMZMYM : ac [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagau [YZYZ : gauucuuuguag] u] gcaagggacuuuuuuu]	-24.00
10	[: [XWXWN : aucaa] g [WMMZMYM : ac [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] auuccugcagcgguuucgcguugaaagaug [ZYZYZNW : cuugu]] gcaagggacuuuuuuu]	-27.60
11	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] a] gcagcgguuucgcguugaaagaug [ZYZ YZNW : cuugu] uucuuuuagcgaagggacuuuuuuu]	-28.30
12	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] a] gcagcgguuucg [XMWN : gaaa] [ZYZY ZNW : cuugu] uucuuuuagcgaagggacuuuuuuu]	-29.40
13	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] a] gcagcgg [NWWXY : [XMWN : gaaa] [ZY ZYZNW : cuugu] uucuuuuag] ggacuuuuuuu]	-31.60
14	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] a] gcagcgg [NWWXY : [XMWN : gaaa] a [Y ZYZNW : cuugu] uucuuuuag] ggacuuuuuuu]	-32.20
15	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : [XXY : augaaacc] ca [ZXXZ : [YXX : uuag] a]] a] gcagcgg [NWWXY : [XMWN : gaaa] a [Y ZYZNW : cuugu] uucuuuuag] ggacuuuuuuu]	-35.30
16	[: [XWXWN : aucaa] gugg [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : [YXX : uuag] a]] a] gcagc [MYWNNX : g [XMWN : gaaa] a [Y ZYZNW : cuugu] uucuuuuagcga] uuuuuu]	-34.60
17	[: [XWXWN : aucaa] g [WY : g [ZYIMZ : c [WYYX : ccgaugaaaccggca [ZXXZ : gccuuagggca]] a] g] gc [MYWNNX : g [XMWN : gaaa] a [YZYZYZNW : cuugu] uucuuuuagcga] uuuuuu]	-30.70

18	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:ccgaugaaaccc[YXX:aacca]uuagggcauggu]a]g]gc[MYWNNX:g[XMWN:gaaa]a [YZYZNW:cuugu]uucuuuuagcgaa]uuuuuu]	-33.70
19	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:ccgaugaaaccc[YXX:aacca]uuagggcauggu]a]g]gc[MYWNNX:g[XYNWMZZYZ: uga[YZYZNW:cuugu]u]aa]uuuuuu]	-41.80
20	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:ccgaugaaaccc[YXX:aacca]uuagggcauggu]a]g]gc[MYWNNX:g[XYNWMZZYZ: ug[YZYZNW:cuugu]]aa]uuuuuu]	-43.50
21	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:ccgaugaaa[XXX:[YXX:aacca]uua]cauggu]a]g]gc[MYWNNX:g[XYNWMZZYZ: ug[YZYZNW:cuugu]]aa]uuuuuu]	-45.80
22	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:ccg[ZWY:aaa[XXX:[YXX:aacca]uua]]ggu]a]g]gc[MYWNNX:g[XYNWMZZYZ: ug[YZYZNW:cuugu]]aa]uuuuuu]	-47.60
23	[: [XWXWN:aucaa]g[WY:g[ZYYMZ:c[WYYX:[XX:g[ZWY:aaa[XXX:[YXX:aacca]uua]]]u]a]g]gc[MYWNNX:g[XYNWMZZYZ: :ug[YZYZNW:cuugu]]aa]uuuuuu]	-51.40

Tabla 4. Mejor plegamiento obtenido para la secuencia de prueba identificada como rb4.

Etapa	Estructura Secundaria	ΔG
0	[:gacucggggugcccuucugcgugaaggcgagaaaauacccguaucaccugaucuggauaaugccagcguaggaagucacggaccaccaggucauu gcuucucacgguuauggcaggagcaaacuaugcaagucgaccuguggucc]	0.00
1	[:gacucg[YYYWM:cccuucugcgugaaggcgagaaa]guaucaccugaucuggauaaugccagcguaggaagucacggaccaccaggucauugc uucucacgguuauggcaggagcaaacuaugcaagucgaccuguggucc]	-4.20
2	[:gacucg[YYYWM:cccuucugcgugaaggcgagaaa]guaucaccugau[XWYY:auaaug]cguaggaagucacggaccaccaggucauugcu ucucacgguuauggcaggagcaaacuaugcaagucgaccuguggucc]	-8.10
3	[:gacucg[YYYWM:cccuucugcgugaaggcgagaaa]guaucaccugau[XWYY:auaaug]cguaggaagucacggaccaccagguca[WWYX WNNXW:ucacgguuauggc]acuaugcaagucgaccuguggucc]	-15.50
4	[:gacucg[YYYWM:cccuucugcgugaaggcgagaaa]guauc[ZXX:ugau[XWYY:auaaug]cguaggaagucacggaccaccagguca[WW YXWNNXW:ucacgguuauggc]acuaugcaagucgaccugucc]cc]	-17.50
5	[:gacucg[YYYWM:cccuucugcgugaaggcgagaaa]guauc[ZXX:ugau[XWYY:auaaug]cguaggaagucacggaccac[XZYWX:a[WWYXWNNXW:ucacgguuauggc]acuaugcaaguc]u]cc]	-23.40
6	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[ZXX:ugau[XWYY:auaaug]cguaggaagucacggaccac[XZYWX:a[WW YXWNNXW:ucacgguuauggc]acuaugcaaguc]u]cc]	-29.70
7	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]gaagucacggaccac[XZYWX:a[WWYXW NNXW:ucacgguuauggc]acuaugcaaguc]uggucc]	-33.70
8	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]gaagucac[YYZXXZ:c[XZYWX:a[WWY XWNNXW:ucacgguuauggc]acuaugcaaguc]]]	-44.80
9	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]gaagucac[YYZXXZ:c[XZYWX:a[WWY XWNNXW:[NXZ:cguua]c]acuaugcaaguc]]]	-47.00
10	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]gaagucac[YYZXXZ:c[XZYWX:a[WWY XWNNXW:[NXZ:cguua]c] [ZXW:augca]c]]]	-49.10
11	[:gacucg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]gaagucac[YYZXXZ:c[XZYWX:au[WY XWNNXW:[NXZ:cguua]c]a [ZXW:augca]c]]]	-49.20
12	[: [YZXW:cg[YYYWM:c[XXWXX:ugcgu]cugagaaa]guauc[XXWM:au[XWYY:auaaug]cg]ga]ac[YYZXXZ:c[XZYWX:au[WYX WNNXW:[NXZ:cguua]c]a [ZXW:augca]c]]]]	-52.90

Tabla 5. Mejor plegamiento obtenido para la secuencia de prueba identificada como rb5.

Etapa	Estructura Secundaria	ΔG
0	[:gggaauuaauaggaaacacucauaaaucgcguggauauggcacgcaaguuuucuaaccgggcaccguaaauguccgacuaugggugagcaauaggaaac cgcacguguacgguuuuuugugauaucagcauugcuugcucuuaauuugagcgggcaauugcuuuuuuauucucauaacggagguagacaggauggau ccacuga]	0.00
1	[:gggaauuaauaggaa [XZXNXZWZ : uaaucgcguggauauggcacgcaaguuuucuaaccgggcaccguaaauguccgac] agcaauggaaccgcac guguacgguuuuuugugauaucagcauugcuugcucuuaauuugagcgggcaauugcuuuuuuauucucauaacggagguagacaggauggauccacu ga]	-6.33
2	[:gggaauuaauaggaa [XZXNXZWZ : uaaucgcguggauauggcacgcaaguuuucuaaccgggcaccguaaauguccgac] agcaauggaaccgcac guguacgguuuuuugugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuuauucucauaacggagguagacaggauggauccacuga]	-29.33
3	[:gggaauuaauaggaa [XZXNXZWZ : uaauc [MXMW : ggauauggcacgcaaguuuucuaaccgggcaccguaa] ccgac] agcaauggaaccgcacg uguacgguuuuuugugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuuauucucauaacggagguagacaggauggauccacuga]	-32.03
4	[:gggaauuaauaggaa [XZXNXZWZ : uaauc [MXMW : ggauauggcacgcaaguuuucuaaccgggcaccguaa] ccgac] agcaauggaaccg [XZX M : uguacgguuuuu] auauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuuauucucauaacggagguagacaggauggauccacuga]	-33.43
5	[:gggaauuaauaggaa [XZXNXZWZ : uaauc [MXMW : ggauauggc [ZX : gcaa] uucuaaccgggcaccguaa] ccgac] agcaauggaaccg [XZ XM : uguacgguuuuu] auauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuuauucucauaacggagguagacaggauggauccacuga]	-34.70
6	[:gggaauuaauaggaa [XZXNXZWZ : ua [ZWX : gcgug] auggc [ZX : gcaa] uucuaaccgggcaccguaaauguccgac] agcaauggaaccg [X ZXM : uguacgguuuuu] auauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuuuauucucauaacggagguagacaggauggauccacuga]	-35.70
7	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : ua [ZWX : gcgug] auggc [ZX : gcaa] uucuaaccgggcaccguaaauguccgac] agcaauggaaccg cacguguacgguuuuuugugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaacggagguagacaggauggauccacuga]	-41.00
8	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : ua [ZWX : gcgug] auggc [ZX : gcaa] uucuaaccgggcaccguaaauguccgac] agcaau [MMZZXX Y : cacgugua] uugugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaacggagguagacaggauggauccacuga]	-46.50
9	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : ua [ZWX : gcgug] auggcacgcaaguuuucuaac [XYMXZ : ccguaaa] ac] agcaau [MMZZXXY : ca cgugua] uugugauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaacggagguagacaggauggauccacuga]	-52.20
10	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : ua [ZWX : gcgug] auggcacgcaaguuuucuaac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZZXXY : cacgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaacggagguagacaggauggauccacuga]	-54.90
11	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : ua [ZWX : gcgug] auggcacgcaaguuuucuaac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZZXXY : cacgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaac [YYZ : gguagacaggaugga] acuga]	-56.80
12	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : uaaucgc [YNY : gauauggca] aaguuuucuaac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZZXXY : cacgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaac [YYZ : gguagacaggaugga] acuga]	-58.70
13	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuucuaac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZZXXY : ca cgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaac [YYZ : gguagacaggaugga] acuga]	-64.40
14	[: [YMYZZWZ : uaauggaa [XZXNXZWZ : uaauc [YXYWY : gauaugg] aaguuuucuaac [XYMXZ : ccguaaa] ac] a [MXZZ : u [MMZZXXY : ca cgugua]] gauauc [ZYXZWWYXNNYXWX : uuuaauuu] uuuuu] auaac [YYZ : gg [NZ : gacagga] ga] acuga]	-65.60

Tabla 6. Mejor plegamiento obtenido para la secuencia de prueba identificada como SplicedLeaderAB.

Etapa	Estructura Secundaria	ΔG
0	[:aacuaaaacaauuuugaagaacaguuucuguaacuuc <u>auugguauguagagacuuc</u>]	0.00
1	[:aacuaaaacaauuuu[WYZZY:aacaguuucugua]uugguauguagagacuuc]	-2.10
2	[:aacuaaa[ZXZ:auuuu[WYZZY:aacaguuucugua]uuggua]agagacuuc]	-2.80
3	[:aacuaaa[ZXZ:auuuu[WYZZY:a[ZXZY:uuu]a]uuggua]agagacuuc]	-8.10
4	[:aacuaaa[ZXZ:[ZNN:uu[WYZZY:a[ZXZY:uuu]a]uu]a]agagacuuc]	-10.10
5	[:aa[XW:aaa[ZXZ:[ZNN:uu[WYZZY:a[ZXZY:uuu]a]uu]a]ag]acuuc]	-10.70

Tabla 7. Mejor plegamiento obtenido para la secuencia de prueba identificada como attenuator.

Etapa	Estructura Secundaria	ΔG
0	[:auccaggaggcuagcgcgugagaagagaaacggaaaacagcgc <u>cugaaagccuccaguggaggcuuuuuug</u>]	0.00
1	[:auccaggaggcuagcg[XYW:gagaagagaa]gaaaacagcgc <u>cugaaagccuccaguggaggcuuuuuug</u>]	-1.00
2	[:auc[XZYY:aggcuagcg[XYW:gagaagagaa]gaaaacagcg]aaagccuccaguggaggcuuuuuug]	-5.80
3	[:auc[XZYY:ag[YXW:agcg[XYW:gagaagagaa]gaaaac]g]aaagccuccaguggaggcuuuuuug]	-8.90
4	[:auc[XZYY:aggcuagcg[XYW:gagaagagaa]gaaaacagcg][ZZYXXWXX:cagu]uuuuug]	-17.90
5	[:auc[XZYY:aggcuagcg[XYW:gagaagagaa]gaaaacagcg]a[ZZYXXWXX:cagu]uuuuug]	-18.10
6	[:auc[XZYY:ag[YXW:agcg[XYW:gagaagagaa]gaaaac]g]a[ZZYXXWXX:cagu]uuuuug]	-21.20

Tabla 9. Mejor plegamiento obtenido para la secuencia de prueba identificada como sbx leader.

Etapa	Estructura Secundaria	ΔG
0	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgaccguaau accauugugaaauggggcgacugcuuuucgcgccgagacugaugucucauaaggcaccggugcuaauuccaucagauugugucugagagagagagag gtaguuguuuacguagaaaagccucuuucucucaugggaaagaggcuuuuugu]	0.00
1	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgaccguaau accauugugaaauggggc [MXZXWYXNWNWX : gcgccgagacugaugucucauaaggcaccggugcuaauuccaucagauugugucugagagauga] uu uacguagaaaagccucuuucucucaugggaaagaggcuuuuugu]	-15.45
2	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgaccguaau accauugugaaauggggc [MXZXWYXNWNWX : gcgccgagacugaugucucauaaggcaccggugcuaauuccaucagauugugucugagagauga] uu uacgua [MZZZZYXXWXXWWXNX : ucau] gu]	-39.55
3	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgac [XYWZZ : uaccauugugaaauggggc [MXZXWYXNWNWX : gcgccgagacugaugucucauaaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZYXXWXXWWXNX : ucau] gu]	-39.65
4	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgac [XYWZZ : uaccauugugaaauggggc [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZYXXWXXWWXNX : ucau] gu]	-45.17
5	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauugggcgagggauuggccuuuugaccccaacagcaaccgac [XYWZZ : uaccuu [YN : gaaauggg] [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZYXXWXXWWXNX : ucau] gu]	-45.27
6	[:aaaugauucaacuuuucaaaaaaauaauaacauuuucucuuuucgagagauuggg [XMZMMY : auuggc] acccaacagcaaccgac [XYWZZ : ua ccuu [YN : gaaauggg] [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZYXXWXXWWXNX : ucau] gu]	-48.57
7	[:aaauga [NNZZZXWNWN : caaaaaauaauaacauuuucucuuuuc] cgaggauuggccuuuugaccccaacagcaaccgac [XYWZZ : uaccuu [YN : gaaauggg] [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZYXXWXXWWXNX : ucau] gu]	-50.57
8	[:aaauga [NNZZZXWNWN : caaaaaauaauaacauuuucucuuuuc] cgaggauuggccuuuugaccccaacagcaaccgac [XYWZZ : ua [XXX WW : guga] ggc [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugugucugagagauga] u] ua [MZZZZY XXWXXWWXNX : ucau] gu]	-58.67
9	[:aaauga [NNZZZXWNWN : caaaaaauaauaacauuuucucuuuuc] cgaggauuggccuuuugaccccaacagcaaccgac [XYWZZ : ua [XXX WW : guga] ggc [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] au [ZZ : ggcaccggugcuaa] caucagauugugucugagagauga] u] ua [MZZZZ YXXWXXWWXNX : ucau] gu]	-58.80
10	[:aaauga [NNZZZXWNWN : caaaaaauaauaacauuuucucuuuuc] cgaggauuggccuuuugaccccaacagcaaccgac [XYWZZ : ua [XXX WW : guga] ggc [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcacc [YM : ugcuaau] caucagauugugucugagagauga] u] ua [MZZZZ YXXWXXWWXNX : ucau] gu]	-61.60
11	[:aaauga [NNZZZXWNWN : caaaaaauaauaacauuuucucuuuuc] cgaggauuggccuuuugaccccaacagcaaccgac [XYWZZ : ua [XXX WW : guga] ggc [MXZXWYXNWNWX : gcgcc [YZYZX : ugau] auaggcaccggugcuaauuccaucagauugug [WX : ugagagau]] u] ua [MZZZZ YXXWXXWWXNX : ucau] gu]	-62.10

12	[:aaauga[NNXZZXWNWN:caaaaaauuaaacaauuuucucuuauc]cgagggauuggccuuuugaccccaacagcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:gcuauuuccaucag]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-71.80
13	[:aaauga[NNXZZXWNWN:caaaaaauuaaacaauuuucucuuauc][XMZMMY:auuggc]acccaacagcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:gcuauuuccaucag]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-74.40
14	[:aaaugau[NXZZXWNWN:caaaaaauuaaacaauuuucucuuauc]g[XMZMMY:auuggc]acccaacagcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:gcuauuuccaucag]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-74.60
15	[:aaaugaucaacuuuucaaaaaauuaaacaauuuucucuuaucgaga[YWWYYY:[XMZMMY:auuggc]ac]agcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:gcuauuuccaucag]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-79.40
16	[:aaaugaucaacuuuucaaaaaauuaaacaauuuucucuuaucgaga[YWWYYY:[XMZMMY:auuggc]ac]agcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:g[XW:aauccauc]]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-80.20
17	[:aaaugaucaacuuuucaaaaaauuaaacaauuu[WXW:uuauc][YWWYYY:[XMZMMY:auuggc]ac]agcaaccgac[XYWZZ:ua[XXZWW:guga]ggc[MXZXWYXNWNWX:gcgcc[YZYZX:ugau]aua[ZYMXXZMMW:g[XW:aauccauc]]gagagauga]u]ua[MZZZZYXXWXXWWXNX:ucau]gu]	-82.40

Tabla 10. Mejor plegamiento obtenido para la secuencia de prueba identificada como thiM leader.

Etapa	Estructura Secundaria	ΔG
0	[:ggaaccaaacgacucggggugcccuucugcguagaaggcugagaaaauaccguaucaccugaucuggauaaugccagcguaggggaagucacggacca ccaggucuuugcuucucacguuauggcaggagcaaacuaugcaagucgaccugcuggaucagcgcaa]	0.00
1	[:ggaaccaaacgacucggggugcccuucugcguagaaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacggac caccaggucuuugcuucucacgu]caggagcaaacuaugcaagucgaccugcuggaucagcgcaa]	-1.14
2	[:ggaaccaaacgacucggggugcccuucugcguagaaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacggac caccaggucuuugcuucucacgu]caggagcaaacuaugcaagucgacc [NYXWY:gauc]caa]	-7.54
3	[:ggaaccaaacgacucggggugcccuuc [WYXMWM:aaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacgg accaccaggucuuugcuucucacgu]caggagcaaac]agucgacc [NYXWY:gauc]caa]	-10.34
4	[:ggaaccaaacgacuc [YY:ggug]cuuc [WYXMWM:aaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacg gaccaccaggucuuugcuucucacgu]caggagcaaac]agucgacc [NYXWY:gauc]caa]	-10.54
5	[:ggaaccaaacgacuc [YY:ggug]cuuc [WYXMWM:aaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacg [YZXX:acca]auugcuucucacgu]caggagcaaac]agucgacc [NYXWY:gauc]caa]	-15.43
6	[:ggaaccaaacgacuc [MYY:gugc]uc [WYXMWM:aaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacg [YZXX:acca]auugcuucucacgu]caggagcaaac]agucgacc [NYXWY:gauc]caa]	-17.83
7	[:ggaaccaaacgacuc [MYY:gugc]uc [WYXMWM:aaggcugagaaaauac [XXMWZ:ucaccugaucuggauaaugccagcguaggggaagucacg [YZXX:acca]auugcuucucacgu]caggagcaaac]agucgaccu [YXWY:gauc]gcaa]	-18.03
8	[:ggaaccaaacgacuc [MYY:gugc]uc [WYXMWM:aaggcugagaaaauaccguaucaccugaucuggauaa [WYXXZ:gcuaggggaagucacg [YZXX:acca]auugcuucucacguua]ggagcaaac]agucgaccu [YXWY:gauc]gcaa]	-23.66
9	[:ggaaccaaacgacuc [MYY:gugc]uc [WYXMWM:aaggcugagaaaauaccguc [ZWX:accu]cuggauaa [WYXXZ:gcuaggggaagucacg [YZXX:acca]auugcuucucacguua]ggagcaaac]agucgaccu [YXWY:gauc]gcaa]	-26.76
10	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aaggcugagaaaauaccguc [ZWX:accu]cuggauaa [WYXXZ:gcuaggggaagucac g [YZXX:acca]auugcuucucacguua]ggagcaaac]]accu [YXWY:gauc]gcaa]	-31.56
11	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aag [YXW:gagaaaauaccguc [ZWX:accu]cuggauaa [WYXXZ:gcuaggggaaguc acg [YZXX:acca]auugcuucucacguua]gg]aac]]accu [YXWY:gauc]gcaa]	-34.96
12	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aag [YXW:gagaaaauaccguc [ZWX:accu]cuggauaa [WYXXZ:gcuag [MMYZZYN :cagc [YZXX:acca]auu]cacguua]gg]aac]]accu [YXWY:gauc]gcaa]	-41.90
13	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aag [YXW:gagaaaauac [XXM:u [ZWX:accu]c]auaa [WYXXZ:gcuag [MMYZZYN :cagc [YZXX:acca]auu]cacguua]gg]aac]]accu [YXWY:gauc]gcaa]	-46.50
14	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aag [YXW:gagaaaauac [XXM:u [ZWX:accu]c]auaa [WYXXZ: [MXYW:a [MMYZZ YN:cagc [YZXX:acca]auu]c]ua]gg]aac]]accu [YXWY:gauc]gcaa]	-52.10
15	[:ggaaccaaaa [XYZXW:c [MYY:gugc]uc [WYXMWM:aag [YXW:gagaaaauac [XXM:u [ZWX:accu]c]auaa [WYXXZ: [MXYW:ag [MYZZ YN:cagc [YZXX:acca]auu]uc]ua]gg]aac]]accu [YXWY:gauc]gcaa]	-53.30

Tabla 11. Mejor plegamiento obtenido para la secuencia de prueba identificada como ms2.

Etapa	Estructura Secundaria	ΔG
0	[: gggugggaccccuuucgggguccugcucaacuuccugucgagcuaaugccauuuuuuuaugucuuuagcgagac]	0.00
1	[: [MYY : ugggaccccuuucgggggu] gcucaacuuccugucgagcuaaugccauuuuuuuaugucuuuagcgagac]	0.00
2	[: [MYY : ugggaccccuuucgggggu] gcucaacuuccugucga [YXWZZ : ugccauuuuuuaugucu] gagac]	-3.60
3	[: [MYY : ugggac [XXXN : uuc] u] gcucaacuuccugucga [YXWZZ : ugccauuuuuuaugucu] gagac]	-7.00
4	[: [MYY : ugggac [XXX : uuucg] u] gcucaacuuccugucga [YXWZZ : ugccauuuuuuaugucu] gagac]	-7.30
5	[: [MYY : ugggac [XXX : uuucg] u] gcucaacuuccu [YWX : ga [YXWZZ : ugccauuuuuuaugucu] ga]]	-10.70
6	[: [MYY : ugggac [XXX : uuucg] u] gcucaacuuccu [YWX : ga [YXWZZ : ugc [XZWW : uuu] ucu] ga]]	-12.60
7	[: [YMYNMYYZ : c [XXX : uuucg]] aacuuccu [YWX : ga [YXWZZ : ugc [XZWW : uuu] ucu] ga]]	-25.40
8	[: [YMYNMYYZXXX : uuuc] aacuuccu [YWX : ga [YXWZZ : ugc [XZWW : uuu] ucu] ga]]	-32.10
9	[: [YMYNMYYZXXX : uuuc] aacuuccu [YWX : ga [YXWZZ : ugc [XZW : uuuua] ucu] ga]]	-32.10

Tabla 12. Mejor plegamiento obtenido para la secuencia de prueba identificada como HDV.

Etapa	Estructura Secundaria	ΔG
0	[: ggaauaggcgggaccccacucugcaggguccgcguuccauccuuuaccugauggcggcaugguccagccuccucgucggcgccggcugggca acaauccgaggggaccgucuccucgguaauaggcgaaugggaccacaaaucucucua]	0.00
1	[: ggaauaggcgggaccccacucugcaggguccgcguuccauccuuuaccugauggcggcaugguccagccuccucgucggcgccg [MXNYMY: caacauuccgagggaccgucc] aauggcgaaugggaccacaaaucucucua]	-3.50
2	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugauggcggcaugguccagccuccucgucggcgccg [MXNYM Y: caacauuccgagggaccgucc] aauggc] ggaccacaaaucucucua]	-3.89
3	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugauggcggcaugguccagccuccucgucggcgccg [MXNYM Y: caacauuccga [YYM: gaccg] c] aauggc] ggaccacaaaucucucua]	-6.59
4	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugauggcggcaugguccagccuc [XN: cgcu] cgccg [MXNY MY: caacauuccga [YYM: gaccg] c] aauggc] ggaccacaaaucucucua]	-7.50
5	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: augguccagccuc [XN: cgcu] cgccg] gc aacauuccga [YYM: gaccg] ccucgguaauaggc] ggaccacaaaucucucua]	-9.90
6	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: augguccagccuccucgucggcgccg] gca acaauuccga [YYM: gaccg] ccucgguaauaggc] ggaccacaaaucucucua]	-12.40
7	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gca acaauuccga [YYM: gaccg] ccucgguaauaggc] ggaccacaaaucucucua]	-14.60
8	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gcaac auuccga [YYM: gaccg] ccucgguaauaggc] ggaccacaaaucucucua]	-21.50
9	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gcaac auu [XXYZYYYY: ccg] uauaggc] ggaccacaaaucucucua]	-35.80
10	[: [MYZ: aug [YXMYM: accccac] aggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gca acaau [XXYZYYYY: ccg] uauaggc] ggaccacaaaucucucua]	-35.90
11	[: [MYZ: aug [YXMYM: accccac] aggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gca acaau [XXYZYYYY: accgu] uauaggc] ggaccacaaaucucucua]	-36.20
12	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gca acaau [XXYZYYYY: accgu] uauaggc] gcaaaaucucucua]	-50.30
13	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: au [YYN: ccca] uccucgucggcgccg] gca a [XZWW: [XXYZYYYY: accgu] u] gc] gcaaaaucucucua]	-54.10
14	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: aug [YNX: ccagccuccucgcu] gccg] gca a [XZWW: [XXYZYYYY: accgu] u] gc] gcaaaaucucucua]	-54.30
15	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: auggc [XXZYX: cuccuc] cgccg] gcaa [X ZWW: [XXYZYYYY: accgu] u] gc] gcaaaaucucucua]	-62.40
16	[: ggaauaggcgggaccccacucugcaggguccg [XMWWX: cauccuuuaccugaugg [XXMYX: a [NYYN: c [XXZYX: cuccuc] c]] gcaa [X ZWW: [XXYZYYYY: accgu] u] gc] gcaaaaucucucua]	-68.80
17	[: [MYZ: aug [YXMYM: accccac] a [YYYWXX: g [XMWWX: cauccuuuaccugaugg [XXMYX: a [NYYN: c [XXZYX: cuccuc] c]] gcaa [XZWW: [XXYZYYYY: accgu] u] gc] gcaaaaucucua]	-69.30

Tabla 13. Mejor plegamiento obtenido para la secuencia de prueba identificada como amv.

Etapa	Estructura Secundaria	ΔG
0	[:aaugcugaccuccacuggguggauuaagggucaagguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgc acauauuaaaugcucaugcaaaacugcaugaagccccuaaggggaugc]	0.00
1	[:aaugcugaccuccacuggguggauuaagggucaagguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgc acauauuaaa [ZWYX :ucaugcaaaacu] gaugccccuaaggggaugc]	-2.40
2	[:aaugcugaccuccacuggguggauuaagggucaagguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgc acauauuaaaugc [WXZWYXZ :aaac] augccccuaaggggaugc]	-8.50
3	[:aaugcugaccuccacuggguggauuaagggucaagguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgc acauauuaaaugc [WXZWYXZ :aaac] aug [XXX :cuaa] augc]	-11.80
4	[:aaugcugacc [WXXZX :ugg] uuaagggucaagguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgcac auauuaaaugc [WXZWYXZ :aaac] aug [XXX :cuaa] augc]	-16.30
5	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] agguaugaaguccuauucgcuccugauaggauagcgcacuucauauugcuuauuaugugcuaacgcacau auuaaaugc [WXZWYXZ :aaac] augccccuaaggggaugc]	-19.10
6	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] a [MMWZWYZZYWX :cuauucgcuccugauaggauagc] gcuuauuaugugcuaacgcacauuaaa gc [WXZWYXZ :aaac] augccccuaaggggaugc]	-29.30
7	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] a [MMWZWYZZYWX :cuauucgcuccugauaggauagc] gc [WWZWZWZYWYX :uaac] augc [WXZWYX Z :aaac] augccccuaaggggaugc]	-45.40
8	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] a [MMWZWYZZYWX :cuauucgcuccugauaggauagc] gc [WWZWZWZYWYX :uaac] augc [WXZWYX Z :aaac] aug [XXX :cuaa] augc]	-48.70
9	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] a [MMWZWYZZYWX : [XWZWN :cgcuccu] gauc] gc [WWZWZWZYWYX :uaac] augc [WXZWYXZ : aaac] aug [XXX :cuaa] augc]	-51.40
10	[:aaugc [WYZXX : [WXXZX :ugg] uuaa] a [MMWZWYZZYWX :cu [ZWNX :gcuccugaua] c] gc [WWZWZWZYWYX :uaac] augc [WXZWYXZ :aaac] aug [XXX :cuaa] augc]	-51.70

ii. Resultados de las pruebas realizadas utilizando secuencias correspondientes a ARN de transferencia (ARNt)

Tabla 15. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Alanina.

Etapa	Estructura Secundaria	ΔG
0	[: ggggaauuagcucaagcggguagagcgcuccuuagcaugcgagagguagcgggaucgacgcccccauucucua]	0.00
1	[: [YYYM: aauuagcucaagcggguagagcgc] uuagcaugcgagagguagcgggaucgacgcccccauucucua]	-4.60
2	[: [YYYM: aauua [YXWX: aagcggua] gc] uuagcaugcgagagguagcgggaucgacgcccccauucucua]	-9.80
3	[: [YYYM: aauua [YXWX: aagcggua] gc] uuagcaugcgagaggua [YXY: ggauca] cccccauucucua]	-13.10
4	[: [MYYYZ: aaua [YXWX: aagcggua] gc] uagcaugcgagaggua [YXY: ggauca] cccccauucucua]	-15.20
5	[: [MYYYZ: aaua [YXWX: aagcggua] gc] uagcaugc [YZM: aggua [YXY: ggauca] cccca] ucua]	-15.50
6	[: [MYYYZ: aaua [YXWX: aagcggua] gc] uagcaugc [YZM: a [YY: ua [YXY: ggauca] c] ca] ucua]	-16.60
7	[: [MYYYZ: aaua [YXWX: aagcggua] gc] uagcaugc [YZM: a [YY: ua [YXY: ggauca] cc] a] ucua]	-20.90
8	[: [MYYYZ: aaua [YXWX: aagcggua] gc] uagcaugc [YZY: [YY: ua [YXY: ggauca] cc] au] ua]	-22.60

Tabla 16. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Arginina.

Etapa	Estructura Secundaria	ΔG
0	[: gggccaguggcgcaauaggauaacgcgucugacuacggaucaagaagauucuagguucgacuccuggcuggcucg]	0.00
1	[: gggccaguggcgcaauaggauaacgcgucugacua [XYMNZ: agaagauucua] acuccuggcuggcucg]	-4.60
2	[: gggccaguggcgcaauaggauaacgcgucugacua [XYMNZ: [ZYZ: agau] a] acuccuggcuggcucg]	-7.20
3	[: [YMYXXZYN: ggcgcaauaggauaacgcgucugacua [XYMNZ: [ZYZ: agau] a] acuccug] g]	-16.86
4	[: [YMYXXZYN: [YMYX: aauggauaac] ugacua [XYMNZ: [ZYZ: agau] a] acuccug] g]	-27.20

Tabla 17. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Asparagina.

Etapa	Estructura Secundaria	ΔG
0	[:gucucuguggcgcaaucggucagagcguucggcuauuaaccgaacggugaguaguaucaagaccaccaggacg]	0.00
1	[:gucucuguggcgcaaucggucagagcguucggcuauuaaccgaacggugaguaguaucaaga [XX: accca] gacg]	-1.00
2	[:gucucuguggcgcaauc [YYWX: agagcguucggcuauuaaccgaacggugaguaguaucaaa] accaggacg]	-3.59
3	[: [YWXNXWY: uggcgcaauc [YYWX: agagcguucggcuauuaaccgaacggugaguaguaucaaa] acc] g]	-9.89
4	[: [YWXNXWY: uggcgcaauc [YYWX: agag [XYWWXYY: cuauuaa] gugaguaguaucaaa] acc] g]	-18.90
5	[: [YWXNXWY: u [YY: cgcaauc [YYWX: agag [XYWWXYY: cuauuaa] gugaguaguaucaaa] a]] g]	-22.20
6	[: [YWXNXWY: u [YY: cgcaauc [YYWX: agag [XYWWXYY: cuauuaa] gu [MZ: guag] caa] a]] g]	-22.80
7	[: [YWXNXWY: u [YY: cgcaauc [YYWX: agag [XYWWXYY: cuauuaa] g [WYZM: uag] a] a]] g]	-26.10

Tabla 18. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Aspartato.

Etapa	Estructura Secundaria	ΔG
0	[:uccuuguuacuauagugguaaguaucucugccugucaugcaugagagagggggucgauuccugacggggag]	0.00
1	[:uccuuguuacuauagugguaagua [WXWX: ugccugucaugcau] gagggggucgauuccugacggggag]	-2.60
2	[:uccuuguuacuauagugguaagua [WXWX: [WYX: cugucau] u] gagggggucgauuccugacggggag]	-6.20
3	[:uccuuguuacuauagugguaagua [WXWX: [WYX: cugucau] u] ga [MYYYM: ucgau] gacggggag]	-14.60
4	[: [WXXNNYWN: acuauagugguaagua [WXWX: [WYX: cugucau] u] ga [MYYYM: ucgau]] g]	-23.00

Tabla 19. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Cisteína.

Etapas	Estructura Secundaria	ΔG
0	[: ggggguagggcucagggauagagcauuugacugcagaucaagaggucgggucgaaucuaaggugccccc]	0.00
1	[: ggggguagggcucagggauagagcauuugacugca [YZNX : aaga] cccggucgaaucuaaggugccccc]	-2.20
2	[: ggggguagggcucagggauagagc [ZWWNYZ : cugca [YZNX : aaga] cccggucgaaucuaaggugccccc]	-4.40
3	[: ggggguagggcucagggauagagc [ZWWNYZ : cugca [YZNX : aaga] cccggucgaaucuaaggugccccc]	-4.90
4	[: ggggguagggcucagggauagagc [ZWWNYZ : [XNY : ca [YZNX : aaga] cc] u] cu [ZYY : ugccc]]	-9.60
5	[: ggggguagg [YXWX : agggaua] [ZWWNYZ : [XNY : ca [YZNX : aaga] cc] u] cu [ZYY : ugccc]]	-12.80
6	[: [YYYYYNZ : gg [YXWX : agggaua] [ZWWNYZ : [XNY : ca [YZNX : aaga] cc] u] cuagg] u]	-26.10
7	[: [YYYYYNZ : gg [YXWX : agggaua] a [WWNYZ : [XNY : ca [YZNX : aaga] cc] u] ucuagg] u]	-27.20

Tabla 20. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Glutamina.

Etapas	Estructura Secundaria	ΔG
0	[: ggcaguaugguagagugguuaagaucaugaacucugaagucagagauacuugaauuugaauugcugguucuguca]	0.00
1	[: [MYXZ : guaugguagagugguuaagaucaugaacucugaagucagagauacuugaauuugaau] gguucuguca]	-0.91
2	[: [YMXZY : uaugguagagugguuaagaucaugaacucugaagucagagauacuugaauuugaauugcugguu] a]	-3.35
3	[: [YMXZY : uaugguaga [MWYMW : uaag] gaacucugaagucagagauacuugaauuugaauugcugguu] a]	-5.37
4	[: [YMXZY : uaugguaga [MWYMW : uaag] gaacucugaag [WXZMZ : gauacuugaau] augcugguu] a]	-9.30
5	[: [YMXZY : ua [NMYNZ : ga [MWYMW : uaag] gaacucugaag [WXZMZ : gauacuugaau] a] guu] a]	-13.00
6	[: [YMXZY : ua [NMYNZ : ga [MWYMW : uaag] gaa [XWXWY : aagu] auacuugaauuugaau] guu] a]	-18.60

Tabla 21. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Ácido glutámico.

Etapa	Estructura Secundaria	ΔG
0	[:ucccugguagucuaguggcuaaaguuuggcgcucucaccgccgggacugguugauuccagaucagggga]	0.00
1	[:ucccugg[WZYNX:uagu]aaguuuggcgcucucaccgccgggacugguugauuccagaucagggga]	-4.30
2	[:ucc[XWYY:[WZYNX:uagu]aaguuuggcgcucucaccgccgggacugguugauu]aucagggga]	-6.03
3	[:ucccugg[WZYNX:uagu]aaguuuggcgcucucaccgccggg[XWYY:uugauu]aucagggga]	-9.50
4	[:ucccugg[WZYNX:uagu]aaguuuggcgcucucaccgccgg[YZ:[XWYY:uugauu]a]agggga]	-10.40
5	[:[NXXXW:gg[WZYNX:uagu]aaguuuggcgcucucaccgccgg[YZ:[XWYY:uugauu]a]]a]	-14.80
6	[:[NXXX:ugg[WZYNX:uagu]aaguuuggcgcucucaccgccgg[YZ:[XWYY:uugauu]a]a]a]	-15.40
7	[:[NXXXWYM:[WZYNX:uagu]aaguuuggcgcucucaccgccggg[XWYY:uugauu]a]a]	-18.30
8	[:[NXXXWY:g[WZYNX:uagu]aaguuuggcgcucucaccgccggg[XWYY:uugauu]au]a]	-18.70
9	[:[NXXXWY:g[WZYNX:uagu]aag[NNNYXY:cucucac]a[XWYY:uugauu]au]a]	-24.60

Tabla 22. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Glicina.

Etapa	Estructura Secundaria	ΔG
0	[:gccuugguggugcagugguagaaauucucgccucccacgugggagaccggguucaauuccggccaugca]	0.00
1	[:gccuugguggugcagugguagaaauucucgccucccacgugggagaccggguucaauucccg [YX:caau] a]	-0.30
2	[:gccuugguggugcagu [YYN:agaaucuc] ucccacgugggagaccggguucaauucccg [YX:caau] a]	-1.20
3	[:gccuugguggugcagu [YYN:agaaucuc] [WXXX:acgu] gaccggguucaauucccg [YX:caau] a]	-5.50
4	[:gccuugguggugcagugguagaaauucucgcc [WXXX:acgu] gaccggguucaauucccg [YX:caau] a]	-5.90
5	[:gccuugguggugcagugguagaaauucucgcc [WXXX:acgu] gacc [YYM:uucaau] cg [YX:caau] a]	-9.70
6	[:gccuugguggugcagugguagaaauucuc [YX:c [WXXX:acgu] gacc [YYM:uucaau] cggccaau] a]	-10.90
7	[: [MXX:uuggu] gcagugguagaaauucuc [YX:c [WXXX:acgu] gacc [YYM:uucaau] cggccaau] a]	-13.30
8	[: [MXX:uuggu] gcagugguagaaauucuc [YX:c [WXXX:acgu] gac [XXYYY:uucaau] ccaau] a]	-18.10
9	[: [MXX:uuggu] gcagugguagaaauucuc [YX: [XWXXX:acgu] ac [XXYYY:uucaau] ccaau] a]	-20.20
10	[: [MXX:uuggu] gcag [WYY:uagaaucucgc [XWXXX:acgu] ac [XXYYY:uucaau]] augca]	-21.20
11	[: [MXX:uuggu] [YXZ:g [WYY:uagaaucucgc [XWXXX:acgu] ac [XXYYY:uucaau]] a] a]	-25.50

Tabla 23. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Histidina.

Etapa	Estructura Secundaria	ΔG
0	[:gcagugacuguauagugguuagcacucuguguuguggccacagcaaccaugguucaaucugagucaugaca]	0.00
1	[:gcagugacuguaua [YWYNNZ :gcacucuguguug] agcaaccaugguucaaucugagucaugaca]	-6.90
2	[:gcagugacuguaua [YWYNNZ : [MXZ :cuc] guug] agcaaccaugguucaaucugagucaugaca]	-7.30
3	[:gcagugacuguaua [YWYNNZ : [MX :acucu] guug] agcaaccaugguucaaucugagucaugaca]	-7.60
4	[:gcagugacuguaua [YWYNNZ : [MXZX :ucu] ug] agcaaccaugguucaaucugagucaugaca]	-10.80
5	[:gcagugacuguaua [YWYNNZ : [MXZX :ucu] ug] agcaac [XZWYMNWXZ :aauc] aca]	-17.70
6	[: [MXZY :uga] aua [YWYNNZ : [MXZX :ucu] ug] agcaac [XZWYMNWXZ :aauc] aca]	-18.90

Tabla 24. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Isoleucina.

Etapa	Estructura Secundaria	ΔG
0	[:ggcugguuaguucaguugguuagagcguggugcuaauaacgcccaaggucguggguucgauccccaauaucggcca]	0.00
1	[:ggcug[MWWZYN:ucaguugguuagagcguggu]aacgcccaaggucguggguucgauccccaauaucggcca]	-2.40
2	[:ggcug[MWWZYN:ucaguugguuagagcguggu]aacgccca[YYNXY:uggguucgauccccaau]a]	-8.40
3	[:ggcug[MWWZYN:[NXZ:guugguuagagcg]u]aacgccca[YYNXY:uggguucgauccccaau]a]	-8.50
4	[:ggcug[MWWZYN:[NXZ:guug[YN:uaga]g]u]aacgccca[YYNXY:uggguucgauccccaau]a]	-8.60
5	[:[YYX:ug[MWWZYN:[NXZ:guug[YN:uaga]g]u]aac]aa[YYNXY:uggguucgauccccaau]a]	-14.50
6	[:[YYX:ug[MWWZYN:[NXZ:guug[YN:uaga]g]u]aac]aa[YYNXY:u[YYY:uucgau]cauau]a]	-18.80
7	[:[YYX:ug[MWWZYN:[NXZ:guug[YN:uaga]g]u]aac]aa[YYNXY:[WYYY:uucgauc]uau]a]	-20.20
8	[:[YYX:ug[MWWZYN:[NXZ:[YNW:gguuag]g]u]aac]aa[YYNXY:[WYYY:uucgauc]uau]a]	-23.40

Tabla 25. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt – Leucina.

Etapa	Estructura Secundaria	ΔG
0	[:gguagcauggcugaguggucuaagauucugaauuaagucuccagucucuugggggugugguuucaaucccaccgcugcua]	0.00
1	[:gguagcauggcugaguggucuaagauucugaauuaa[YNXNXXZM:ucucu]gugguuucaaucccaccgcugcua]	-11.00
2	[:gguagcauggcugaguggucuaa[YZWW:cugaauuaa[YNXNXXZM:ucucu]gugguuuc]ccaccgcugcua]	-11.80
3	[:[MYNZYX:auggcugaguggucuaa[YZWW:cugaauuaa[YNXNXXZM:ucucu]gugguuuc]ccacc]a]	-15.70
4	[:[MYNZYX:auggcugaguggucuaa[YZWW:cu[MZZNNZ:a[YNXNXXZM:ucucu]g]uc]ccacc]a]	-20.50
5	[:[MYNZYX:au[YMKN:gagu]uaa[YZWW:cu[MZZNNZ:a[YNXNXXZM:ucucu]g]uc]ccacc]a]	-27.70

Tabla 26. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Lisina.

Etapa	Estructura Secundaria	ΔG
0	[: aaccgaauagcuuaguugaugaagcgugagacucuuaaaucucagggguaguggguucaagccccacauuggaca]	0.00
1	[: aaccgaauagcuuaguugaugaagcgugagacucuuaaaucucagggguagugg [YWNXZZ : gcccaca] a]	-5.20
2	[: aaccgaaua [YXWW : aguugaug] gugagacucuuaaaucucagggguagugg [YWNXZZ : gcccaca] a]	-8.50
3	[: aaccgaaua [YXWW : aguugaug] gugag [ZXNXW : uaaucuc] agugg [YWNXZZ : gcccaca] a]	-11.20
4	[: aaccgaaua [YXWW : aguugaug] gugag [ZXNXW : uaaucuc] a [YWYYY : uucaagc] auuggaca]	-13.70
5	[: aaccgaaua [YXWW : aguugaug] g [WYZYZ : cucuuaa] gggua [YWYYY : uucaagc] auuggaca]	-14.70
6	[: aa [XXMZ : aua [YXWW : aguugaug] g [WYZYZ : cucuuaa] gggua [YWYYY : uucaagc] a] aca]	-17.60

Tabla 27. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Metionina.

Etapa	Estructura Secundaria	ΔG
0	[: agcagaguggugcaguggaagcauaccuaugggcccauaaaccagagguugauggauggaaaccauccucugcua]	0.00
1	[: agcagaguggugcaguggaagcauaccua [WYYY : cccauaa] gagguugauggauggauggaaaccauccucugcua]	-4.50
2	[: agcagaguggugcaguggaagcauaccua [WYYY : cccauaa] ga [YYWW : gauggaugga] auccucugcua]	-7.70
3	[: agcagagug [MWYX : aguggaa] accua [WYYY : cccauaa] ga [YYWW : gauggaugga] auccucugcua]	-10.80
4	[: agcagagug [MWYX : aguggaa] [ZXXW : a [WYYY : cccauaa] g] ugauggauggauggauggaaaccauccucugcua]	-11.80
5	[: agcagagug [MWYX : aguggaa] [ZXXW : a [WYYY : cccauaa] g] ugau [YYZWYY : aaa] ucugcua]	-19.50
6	[: [ZYXZYZ : gug [MWYX : aguggaa] [ZXXW : a [WYYY : cccauaa] g] ugau [YYZWYY : aaa]] a]	-25.90

Tabla 28. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Fenilalanina.

Etapa	Estructura Secundaria	ΔG
0	[:gccgaaauagcucaauugggagaguguuagacugaagaucucugcaggucucugguucaauuccggguuucgaca]	0.00
1	[:gccgaaauagcucaauugggagaguguuagacuga [ZYZNXW:ucugc] cugguucaauuccggguuucgaca]	-3.90
2	[:gccgaaauagcucaauugggagagugua [MZXNM:a [ZYZNXW:ucugc] c] caauuccggguuucgaca]	-7.50
3	[:gccgaaauagcucaauugggagagugua [MZXN:ga [ZYZNXW:ucugc] cu] caauuccggguuucgaca]	-8.70
4	[:gc [XYZZZN:agcucaauugggagagugua [MZXN:ga [ZYZNXW:ucugc] cu] caauuccgg] aca]	-11.20
5	[:gc [XYZZZN:ag [XNX:aaau] agagugua [MZXN:ga [ZYZNXW:ucugc] cu] caauuccgg] aca]	-15.20
6	[:gc [XYZZZN:ag [XNX:aaau] a [YZMW:guua [MZXN:ga [ZYZNXW:ucugc] cu] ca] cgg] aca]	-17.20

Tabla 29. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Prolina.

Etapa	Estructura Secundaria	ΔG
0	[:ggcucguuggucuagggguaugauucucgcuuagggugcgagaggucccgguucaaaaucccgagagccc]	0.00
1	[:ggcucguuggucuagggguauga [NWXWXYX:uuaggg] gucccggguucaaaaucccgagagccc]	-10.00
2	[:ggcucguuggucuag [YYY:uauga [NWXWXYX:uuaggg] gucccggguucaaaa] ggacgagccc]	-12.50
3	[: [YXWXYWN:ggucuag [YYY:uauga [NWXWXYX:uuaggg] gucccggguucaaaa] g] c]	-22.70
4	[: [YXWXYWN:ggucuag [YYY:uauga [NWXWXYX:uuaggg] gu] gggguucaaaaucccg] c]	-24.30
5	[: [YXWXYWN:ggucuag [YYY:uauga [NWXWXYX:uuaggg] gu] [YMW:ucaaa] cg] c]	-29.00
6	[: [YXWXYWN:ggucuag [YYY:uauga [NWXWXYX:uuaggg] gu] [YM:uucaaaa] cg] c]	-30.00
7	[: [YXWXYWN:ggucuag [YYY:uauga [NWXWXYX:uuaggg] gu] [YYY:uucaaaa] g] c]	-30.80

Tabla 30. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Serina.

Etapa	Estructura Secundaria	ΔG
0	[:ugaguuguagcugagugguuaaggcaacgagcuagaaaucguugguuucucucugugcagguuugaauccugcuaauuaug]	0.00
1	[:ugaguuguagcugagugguuaaggcaa [XMZMXN : agaaaucguugguuucucucugugca] aaucugcuaauuaug]	-2.50
2	[:ugaguug [WZMXN : gagu] aggcaa [XMZMXN : agaaaucguugguuucucucugugca] aaucugcuaauuaug]	-4.60
3	[:uga [YNN : g [WZMXN : gagu] a] aa [XMZMXN : agaaaucguugguuucucucugugca] aaucugcuaauuaug]	-7.30
4	[:uga [YNN : g [WZMXN : gagu] a] aa [XMZMXN : [ZYZZZN : cguu] cucugugca] aaucugcuaauuaug]	-11.10
5	[:uga [YNN : g [WZMXN : gagu] a] aacgagcu [ZYZZZN : cguu] cucugu [YXZYY : uuugaau] uaauuaug]	-14.90
6	[:uga [YN : ug [WZMXN : gagu] ag] aacgagcu [ZYZZZN : cguu] cucugu [YXZYY : uuugaau] uaauuaug]	-15.60
7	[:uga [YN : ug [WZMXN : gagu] ag] aacgagcu [ZYZZZN : ucuug] cucugu [YXZYY : uuugaau] uaauuaug]	-15.90
8	[:uga [YN : ug [WZMXN : gagu] ag] aac [YZY : cu [ZYZZZN : ucuug]] ugu [YXZYY : uuugaau] uaauuaug]	-18.80

Tabla 31. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Treonina.

Etapa	Estructura Secundaria	ΔG
0	[: agcaccauggcucuagcugguuaaagcaccugucuaguuaaacaggagauccugaguuucaauuccauggugccu]	0.00
1	[: agcaccauggcucuagcu [YYW : uaaagc] ugucuaguuaaacaggagauccugaguuucaauuccauggugccu]	-1.40
2	[: a [YXZXXZW : ggcucuagcu [YYW : uaaagc] ugucuaguuaaacaggagauccugaguuucaauucca] cu]	-9.61
3	[: a [YXZXXZW : ggcucuagcu [YYW : uaaagc] ug [NXW : aguaaac] agauccugaguuucaauucca] cu]	-11.00
4	[: a [YXZXXZW : ggcucuagcu [YYW : uaaagc] ug [WXN : aguaaca] gauccugaguuucaauucca] cu]	-12.40
5	[: a [YXZXXZW : ggcucuagcu [YYW : uaaagc] ugucuaguuaaacaggagauccu [YZMW : uuca] ca] cu]	-12.40
6	[: a [YXZXXZW : ggc [NWZY : cu [YYW : uaaagc] ugu] uaaacaggagauccu [YZMW : uuca] ca] cu]	-13.80
7	[: a [YXZXXZW : g [MXW : uagcu [YYW : uaaagc] ugucu] aaacaggagauccu [YZMW : uuca] ca] cu]	-14.20
8	[: a [YXZXXZW : g [MXW : uagcu [YYW : uaaagc] ugucu] aaa [XZYY : agau] aguuucaauucca] cu]	-19.90
9	[: a [YXZXXZW : g [MXW : ua [MX : u [YYW : uaaagc] u] cu] aaa [XZYY : agau] aguuucaauucca] cu]	-22.30

Tabla 32. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Triptófano.

Etapa	Estructura Secundaria	ΔG
0	[: ggaaggauggggccaagcuggaaagccugugggcuccacagucaugugccuggguucaauucccaguucugcau]	0.00
1	[: ggaagga [WYYMYXX : aagcuggaaagccugug] cagucaugugccuggguucaauucccaguucugcau]	-10.30
2	[: ggaagga [WYYMYXX : aa [YXW : ggaa] cugug] cagucaugugccuggguucaauucccaguucugcau]	-16.20
3	[: ggaagga [WYYMYXX : aa [YXW : ggaa] cugug] cagucaugugc [XWYYY : uucauu] uucugcau]	-23.50
4	[: g [YZ : agga [WYYMYXX : aa [YXW : ggaa] cugug] cag] augugc [XWYYY : uucauu] uucugcau]	-24.30
5	[: g [YZ : agga [WYYMYXX : aa [YXW : ggaa] cugug] cag] au [MWYX : [XWYYY : uucauu] uucu]]	-26.40

Tabla 33. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Glutamina.

Etapas	Estructura Secundaria	ΔG
0	[:ccuucuuaguuacagcugguagagcagaggacuauagcuacuuccucaguaggagacguccuuagguugcugguucgauuccagcuugaagga]	0.00
1	[:ccuucuuaguu [XZMXN : gguagagcagaggacuauagcuacuuccucaguaggagacguccuua] cugguucgauuccagcuugaagga]	-0.69
2	[: [XXWWXZZ : uaguu [XZMXN : gguagagcagaggacuauagcuacuuccucaguaggagacguccuua] cugguucgauuccagc] a]	-5.99
3	[: [XXWWXZZ : uaguu [XZMXN : gguagagca [YZYYZ : cuauagcuacu] aguaggagacguccuua] cugguucgauuccagc] a]	-10.80
4	[: [XXWWXZZ : uaguu [XZMXN : gguagagca [MZYYZX : uauagcuacuuccucaguaggagac] a] cugguucgauuccagc] a]	-11.40
5	[: [XXWWXZZ : ua [YNWX : agcuggua] a [MZYYZX : uauagcuacuuccucaguaggagac] agguugcugguucgauuccagc] a]	-17.10
6	[: [XXWWXZZ : ua [YNWX : agcuggua] a [MZYYZX : uaua [MXW : acuuccuc] aggagac] agguugcugguucgauuccagc] a]	-17.70
7	[: [XXWWXZZ : ua [YNWX : agcuggua] a [MZYYZX : uaua [MXW : acuuccuc] aggagac] agguu [YXWYY : uucgauu]] a]	-23.70
8	[: [XXWWXZ : aua [YNWX : agcuggua] a [MZYYZX : uaua [MXW : acuuccuc] aggagac] agguu [YXWYY : uucgauu] u] a]	-24.20
9	[: [XXWWXZ : aua [YNWX : agcuggua] a [MZYYZX : uauag [XWZXW : uccuc] gagac] agguu [YXWYY : uucgauu] u] a]	-29.40

Tabla 2. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Valina.

Etapas	Estructura Secundaria	ΔG
0	[:gcuucuguaaanguagugguaucauacauucgccucacacaugaaaggucaccaguuugagaccgggcaaaaaca]	0.00
1	[:gcuucuguaaanguag [WYYNN : aucauucgccucacacaugaaaggucaccaguuugagaccg] aaaca]	-1.63
2	[:gcuucuguaa [WYW : ag [WYYNN : aucauucgccucacacaugaaaggucaccaguuugagaccg] aa]]	-3.73
3	[:gcuucuguaa [WYW : ag [WYYNN : aucauucgccucacacaugaaa [YYWX : accaguuuga] g] aa]]	-6.70
4	[:gcuucuguaa [WYW : ag [WYYNN : aucauucgcc [WXZ : caca] aa [YYWX : accaguuuga] g] aa]]	-10.80

Tabla 2. Mejor plegamiento obtenido para la secuencia correspondiente al ARNt - Selenocisteína.

Etapa	Estructura Secundaria	ΔG
0	[:gaccguguggccuuaauggauaaggugucugacuucagaucagaagauugaggguuugagucccuugugguca]	0.00
1	[:gaccgugug[MXXWWZ:augga]gucugacuucagaucagaagauugaggguuugagucccuugugguca]	-5.40
2	[:gaccgugug[MXXWWZ:augga][MWXWY:acuu]cagaagauugaggguuugagucccuugugguca]	-8.80
3	[:gaccgugug[MXXWWZ:augga]g[WXWYZ:cuucaga]agauugaggguuugagucccuugugguca]	-8.90
4	[:[YZXXMNM:ug[MXXWWZ:augga]g[WXWYZ:cuucaga]agauugaggguuugagucccuu]a]	-14.00
5	[:[YZXXMNM:ug[MXXWWZ:augga]g[WXWYZ:cuucaga]agauu[MZYYY:uuugagu]]a]	-18.70
6	[:[YZXXMN:gug[MXXWWZ:augga]g[WXWYZ:cuucaga]agauu[MZYYY:uuugagu]u]a]	-19.30
7	[:[YZXXM:ugug[MXXWWZ:augga]g[WXWYZ:cuucaga]agauu[MZYYY:uuugagu]ug]a]	-20.60