Desarrollo de una biblioteca de extracción de características de datos biológicos secuenciales

Cristian Ariel Yones

PROYECTO FINAL DE CARRERA
INGENIERÍA EN INFORMÁTICA
UNIVERSIDAD NACIONAL DEL LITORAL

DIRECTOR
Diego H. Milone

1 de abril de 2014

Agradecimientos

Agradezco en primer lugar a mi director Diego Milone y a Georgina Stegmayer por su constante y paciente seguimiento, por orientarme y motivarme durante la realización de este trabajo.

A la cátedra de Proyecto Final de Carrera por su buena predisposición y sus valiosos consejos.

A todos los docentes de la Facultad de Ingeniería y Ciencias Hídricas que compartieron sus conocimientos, dentro y fuera de clase, haciendo que mi formación profesional se resuma en satisfacciones académicas y personales.

A mis compañeros y amigos de la Facultad que me ayudaron a transitar este camino, de los cuales aprendí muchas cosas que van mas allá de las relacionadas con el estudio.

Principalmente le agradezco con todo mi cariño a las personas que hicieron y hacen todo para que yo pueda alcanzar mis metas, a mi familia.

Índice general

Resumen

Los miRNAs son una clase abundante de diminutas moléculas que regulan la expresión de genes codificantes de proteínas en plantas y animales. En el presente trabajo se realizó el estudio y codificación de las características más utilizados para la clasificación de secuencias de miRNA. Se encontraron en total 83 características en trabajos científicos de los últimos 12 años, de las cuales se seleccionaron 73 para incluir sus procesos de extracción en una biblioteca de funciones de Matlab®. De éstas, 67 procesos de extracción fueron codificados internamente y para el resto se creó una interfaz con software externo que realiza estos procesos. Además, se desarrolló una interfaz gráfica de usuario que permite acceder a las funciones de la biblioteca, aún sin tener conocimientos de programación. Esta interfaz además provee las siguientes funcionalidades: el procesado por lotes, el manejo de archivos de configuración, la carga de secuencias desde archivos en formato estándar en bioinformática, la escritura de resultados en formato de texto separado por comas y la presentación de informes con los resultados.

Se seleccionaron secuencias de miRNA reales y se analizaron con distintos programas de otros autores, para luego comparar los resultados con los obtenidos por la biblioteca y de esta forma comprobar su correcto funcionamiento. Para otros algoritmos de extracción más simples se analizaron las secuencias manualmente y se calcularon los resultados esperados para luego compararlos con los obtenidos por la biblioteca. Se realizaron también pruebas de desempeño a las funciones de extracción de características, obteniendo tiempos en el orden de los milisegundos para la mayoría de las funciones. Las funciones que requerían más de un segundo en analizar una secuencia fueron paralelizadas, disminuyendo los tiempos con una eficiencia del 85 % en promedio.

Capítulo 1

Introducción

Después de una breve introducción al tema, en este capítulo se presenta la motivación que impulsó la realización del Proyecto Final de Carrera, cuáles son los objetivos perseguidos y cuál es su alcance.

1.1. Motivación

En los últimos años, la biología molecular demostró un importante avance en distintas áreas de la investigación. Esta disciplina científica estudia la estructura, función y composición de las moléculas biológicamente importantes y se relaciona fuertemente con otras áreas como la genética y la bioquímica. En la actualidad se invierte un gran esfuerzo en estudiar las interacciones entre los numerosos sistemas de las células, incluyendo las interacciones entre los diferentes tipos de ácido desoxirribonucleico (DNA), ácido ribonucleico (RNA) y los procesos de síntesis de proteínas, y también cómo estas interacciones son reguladas. Estos conocimientos son utilizados para crear nuevas drogas, diagnosticar enfermedades, mejorar genéticamente animales y plantas, entre otras aplicaciones [2, 33].

Los avances en este campo permiten la generación de una gran cantidad de información que requiere el uso de herramientas de cálculo altamente especializadas para el análisis. En este contexto, la bioinformática ha tomado un papel muy importante ya que aporta herramientas que posibilitan la explotación de estos datos. Ésta se ocupa de "la investigación, desarrollo o aplicación de herramientas computacionales y aproximaciones para la expansión del uso de datos biológicos, médicos, conductuales o de salud, incluyendo aquellas herramientas que sirvan para adquirir, almacenar, organizar, analizar o visualizar tales datos"[23].

El análisis de datos biológicos con estructura secuencial ha sido de gran

relevancia para el desarrollo de la bioinformática en las últimas décadas [26, 8]. Entre las diversas secuencias de este área se pueden citar especialmente las de nucleótidos y las de aminoácidos, que han dado lugar a importantes desarrollos en modelado estadístico e informática en general. En particular, el RNA es una secuencia que se desempeña como intermediaria entre el DNA y la síntesis de proteínas, permitiendo que en definitiva se exprese la información contenida en los genes de un organismo. Recientemente se ha descubierto un nuevo tipo de pequeñas moléculas de RNA, denominadas microRNA o miRNA, que regulan la expresión de los genes y se encuentran presentes tanto en animales como en plantas. Su importancia en procesos biológicos clave ha sido ampliamente documentada [27, 1], como en los casos del desarrollo y diferenciación de las células y en el metabolismo. Estudios recientes demuestran que los miRNAs estarían implicados, por ejemplo, en la evolución del cáncer (sea como inhibidores o promotores de este) [9] y en procesos de infección viral [18].

En este contexto, se ha potenciado últimamente el surgimiento de técnicas computacionales para la clasificación de miRNAs directamente a partir de las características de la secuencia de RNA. Estas técnicas se pueden clasificar en tres grandes categorías: i) enfoques experimentales dirigidos por datos, mediante secuenciado y clonación directa; ii) métodos comparativos basados en la conservación ya sea de la secuencia o de la estructura entre distintas especies; y iii) métodos basados en aprendizaje de máquina, a partir de características inherentes a la secuencia y la estructura secundaria de las moléculas de miRNA, intentando encontrar nuevos miRNAs que sean de algún modo parecidos a aquellos que ya se conocen [20]. Los primeros métodos basados en aprendizaje de máquina han usado representaciones simples de tipo "bag of words" para extraer las características estructurales principales de los miRNA conocidos y generar artificialmente ejemplos negativos [37]. Con estos conjuntos se entrena luego un clasificador binario para la identificación de secuencias candidatas a miRNA, usando en la gran mayoría de los casos máquinas de vectores de soporte (SVM) [20, 15].

En la actualidad existe una gran variedad de características que se pueden extraer de la secuencia de RNA o de su estructura secundaria: la estructura tipo tallo-lazo, la frecuencia de aparición de los nucleótidos, el número de bases apareadas en el plegado, la energía mínima libre de plegado y la frecuencia de tripletas son algunas de éstas [37, 15]. La gran cantidad de características propuestas en la bibliografía y la diversidad de herramientas para extraerlas son un problema que dificulta su aplicación. Éstas últimas a veces son de difícil acceso y su utilización requiere pasar por un proceso de aprendizaje. Otras veces es necesario codificar los procesos de extracción, lo que requiere conocimientos de programación y la duplicación de trabajo ya

realizado por otros autores.

En este contexto se planteó el presente proyecto final de carrera para crear una interfaz unificada de acceso a todos estos procesos de extracción de características. Esta interfaz consiste en una biblioteca que implementa algunos de estos procesos de extracción internamente y que hace llamadas a herramientas externas para brindar funciones cuya implementación no fue factible en el contexto de este trabajo. También se propuso desarrollar una interfaz gráfica de usuario que permita acceder a las funcionalidades de la biblioteca. Ésta posibilitaría que la biblioteca pueda ser aprovechada por usuarios que no tienen conocimientos de programación. Este es el caso de muchos biólogos y profesionales de la salud que pueden estar interesados en las funciones que la biblioteca brinda. La interfaz gráfica además permite la lectura y escritura de datos en formatos de archivo compatibles a los usados en la actualidad en el campo de la bioinformática y brinda al usuario informes de los resultados que permiten realizar un análisis de los vectores de características, previo a la etapa de clasificación automática.

Se considera que el desarrollo de esta biblioteca y la interfaz de usuario que facilita su utilización es un paso fundamental y necesario para el posterior desarrollo de mejores métodos de clasificación de secuencias de RNA. Éstas facilitarían el estudio de la utilidad y la relevancia de las características utilizadas, además de permitir la experimentación con distintos tipos de clasificadores y algoritmos de entrenamiento. El proyecto se llevó a cabo en el Centro de Investigación en Señales, Sistemas e Inteligencia Computacional (**sinc**(i)) y, siendo esta problemática una de las líneas de investigación en las que trabaja este centro, se considera que este proyecto puede colaborar con el trabajo que se está realizando. Por otro lado, realizar este proyecto fue una buena oportunidad para que el autor inicie los estudios en el tema.

1.2. Objetivos

1.2.1. Generales

- Desarrollar una biblioteca de extracción de características de secuencias biológicas.
- Desarrollar una interfaz gráfica de usuario que facilite el uso de la biblioteca.

1.2.2. Específicos

- Realizar un estudio sobre las características que se utilizan con mayor frecuencia en el estado del arte para la clasificación de secuencias de microRNA.
- Diseñar la biblioteca y codificar los algoritmos de extracción de características y las llamadas a funciones externas que serán parte de esta.
- Realizar pruebas de desempeño de la biblioteca para tener datos sobre el costo computacional de la extracción de cada característica.
- Codificar funciones secundarias de la biblioteca, tales como las necesarias para guardar y abrir secuencias RNA de archivos y las que permitan el procesado por lotes de los datos.
- Desarrollar los distintos módulos de la interfaz gráfica: uno de acceso a las funcionalidades de la biblioteca, otro de presentación de informes sobre los vectores de características extraídas.

1.2.3. Alcance

Se hizo uso de software libre en los casos en que el proceso de extracción de alguna característica era demasiado complejo de codificar. En los casos donde no se encontró software que implemente esa función, se dejó fuera de la biblioteca la característica. Sólo se implementó la función de guardar y abrir en formatos de archivos libres y de amplia utilización en el campo del aprendizaje de máquina y más puntualmente de la problemática tratada.

1.3. Estructura

Capítulo 2: En el capítulo 2 se pretende dar al lector el marco teórico necesario para poder comprender el desarrollo del proyecto. En primer lugar se hace una breve introducción a las cuestiones biológicas: la estructura y funcionamiento de los ácidos nucleicos, el microRNA y sus particularidades. Por último se hace un repaso del estado del arte en cuanto a la identificación de estas moléculas, haciendo hincapié en las características que se utilizan.

Capítulo 3: En el tercer capítulo se presenta la biblioteca desarrollada y la interfaz gráfica de usuario. En primer lugar se explican cuáles fueron las tecnologías utilizadas en el desarrollo: el lenguaje de programación, bibliotecas, software externo utilizado para realizar cálculos complejos, etc. Luego se

describe la estructura de la biblioteca y su utilización. Por último se presenta la interfaz gráfica desarrollada y su funcionamiento.

Capítulo 4: En este capítulo se describe la metodología de las pruebas realizadas a las funciones de extracción de características de la biblioteca y sus resultados. Además se presenta un resumen de los cambios y correcciones realizadas durante este proceso.

Capítulo 5: En este capítulo se detallan los objetivos cumplidos y los trabajos futuros.

Capítulo 2

Marco teórico

El presente capítulo pretende dar al lector el marco teórico necesario para poder comprender el desarrollo del proyecto. En primer lugar se hará una breve introducción a las cuestiones biológicas: la estructura y funcionamiento de los ácidos nucleicos, el microRNA y sus particularidades. Por último se hará un repaso del estado del arte en cuanto a la clasificación de estas moléculas, haciendo hincapié en las características que se utilizan.

2.1. DNA, RNA y microRNA

El ácido desoxirribonucleico, abreviado como DNA por su nombre en ingles, es una molécula que codifica instrucciones genéticas utilizadas en el desarrollo y funcionamiento de todos los seres vivos conocidos y de algunos virus. La mayoría de las moléculas de DNA son bicatenarias, es decir, se forman por dos largas cadenas compuestas por elementos más simples conectados entre sí llamados nucleótidos. Cada nucleótido, a su vez, está formado por un azúcar (la desoxirribosa), un grupo fosfato que actúa como enganche entre cada nucleótido con el siguiente y una base nitrogenada que puede ser adenina (A), timina (T), citosina (C) o guanina (G). Como lo que distingue a un nucleótido de otro es la base nitrogenada, la secuencia del DNA se especifica nombrando sólo la secuencia de sus bases.

Cada base tiene una complementaria con la que se atrae y pueden formar enlaces (A con T y C con G), que son los que mantienen unidas las dos hebras de la cadena de DNA. Las dos hebras están unidas entre sí por unas conexiones llamadas puentes de hidrógeno. La disposición secuencial de estas cuatro bases a lo largo de la cadena es la que codifica la información genética.

Para que la información que contiene el DNA pueda ser expresada, debe copiarse en primer lugar en unas secuencias más cortas llamados RNA. Éstas

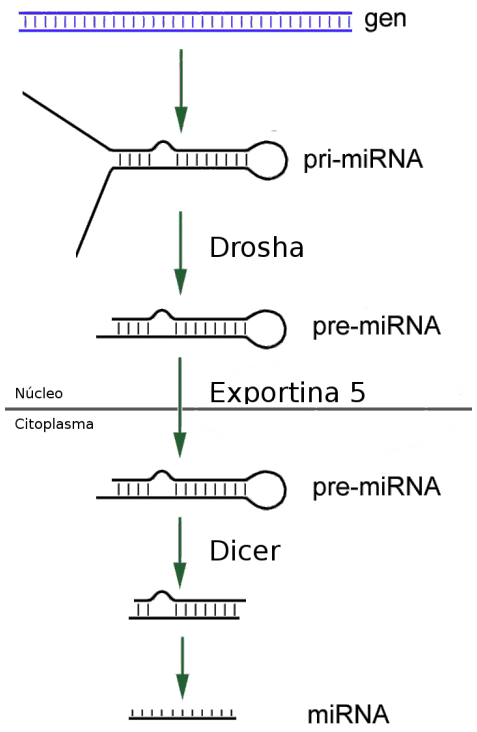


Figura 2.1: proceso de formación de una molécula de microRNA

están compuestas por nucleótidos como el DNA, con la única diferencia que la base timina se convierte en uracilo (U) y la desoxirribosa (el azúcar) en ribosa. Esta nueva base se atrae con la adenina tal como lo hacia la timina, y además se atrae con menor fuerza con la guanina, logrando establecer enlaces en algunos casos. Este proceso de copiado de DNA a RNA se denomina transcripción.

Una vez procesadas en el núcleo celular, las moléculas de RNA pueden salir al citoplasma para su utilización posterior. La información genética se halla codificada en las secuencias de nucleótidos y especifica la secuencia de los aminoácidos que deben tener las proteínas que se sintetizarán [6].

Los microRNA (abreviado como miRNA) son moléculas de RNA monocatenario de una longitud de 21 a 25 nucleótidos, transcritas a partir de DNA pero que no son traducidas a proteínas sino que su tarea es regular la expresión génica mediante diversos procesos. Estas moléculas fueron descritas inicialmente por [19], aunque el término "microRNA" apareció más tarde.

Se ha demostrado que los miRNA participan en muchos procesos biológicos importantes, como en la formación de órganos, la hematopoyesis o la apoptosis. Por ejemplo el miR-122, que se expresa específicamente en el híga-

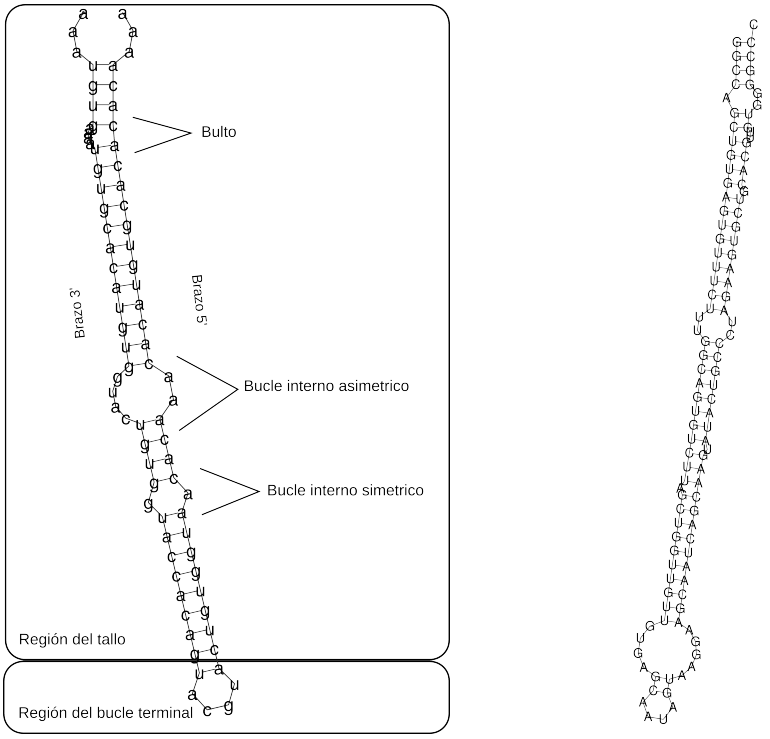
do, es necesario para que el virus de la hepatitis C se exprese de manera eficiente [17]. Algunos recuentos en humanos identificaban hasta 800 miRNA, lo que implicaría que estas moléculas podrían representar como mínimo el 3 % de todos los genes humanos [3].

La secuencia de DNA que codifica para un gen de miRNA tiene una longitud que supera al tamaño final del propio miRNA (60 a 70 nucleótidos) e incluye la región del miRNA y una región complementaria que permite su apareamiento. Esto genera que durante la transcripción de esta secuencia de DNA, se formen regiones que se pliegan formando una horquilla y generan un RNA bicatenario primario de mayor longitud, conocido como estructura secundaria o pri-miRNA. Posteriormente, una enzima nuclear llamada *drosha* corta las bases de la horquilla, formando lo que se denomina miRNA precursor o pre-miRNA (ver Figura 2.1). Éste es transportado desde el núcleo al citoplasma por la *exportina 5*. Una vez que el pre-miRNA está en el citoplasma es fragmentado por la enzima *dicer*, que lo corta hasta la longitud de 21-23 nucleótidos (nt). Finalmente se separan los brazos de la molécula, uno de ellos se disuelve y el otro se convierte en una molécula de miRNA.

El precursor de un miRNA tiene una estructura que se puede dividir en dos regiones: la región del tallo y la región del bucle terminal, o simplemente bucle. En la Figura 2.2a se puede ver un ejemplo de estructura secundaria con sus distintas partes. Además, en esta imagen se puede ver un bulto que aparece cuando algunos nucleótidos de un brazo no se emparejan y dos bucles que aparece cuando los nucleótidos de ambos brazos no se emparejan. Si la cantidad de nucleótidos no emparejados es igual, el bucle es simétrico, de otra forma es asimétrico. En la Figura 2.2b se muestra un ejemplo real: el pri-miRNA hsa-mir-34a. Las letras en ambas figuras indican el tipo de base que forma cada nucleótido.

Debido a las dificultades que conlleva detectar miRNAs del genoma utilizando técnicas experimentales, los métodos computacionales juegan un rol muy importante en la clasificación de miRNAs [21]. Recientemente muchos métodos basados en aprendizaje de máquina fueron presentados para distinguir pre-miRNA real de otras secuencias que forman estructuras secundarias tipo horquilla, pero que no contienen miRNA. Estos métodos extraen distintos tipos de características, tanto de la secuencia primaria como de la estructura secundaria, para luego realizar la clasificación automática.

La aparición de una secuencia en distintas especies de animales o plantas es un buen indicador de la presencia de miRNA. Es muy común que se tomen grupos de secuencias, se alineen para encontrar la posición donde se maximicen las coincidencias y se analicen en conjunto. En el capítulo siguiente se analizarán todas estas cuestiones con más detalle.



(a) (b)

Figura 2.2: estructuras secundarias tipo tallo horquilla; (a) ejemplo con regiones delimitadas; (b) miRNA hsa-mir-34a (humano)

2.2. Extracción de características

A continuación se presenta una lista detallada de características utilizadas en trabajos científicos en los últimos 10 años. Todos los procesos de extracción de las características que se listan fueron codificados en la biblioteca (o las llamadas al software externo que lo realiza). En el Anexo A se listan otras características encontradas en la bibliografía que no se han implementado por estar fuera del alcance de este trabajo final de carrera debido a su complejidad. Para la organización del documento, las características se separaron en distintos grupos según el tipo de información que extraen de la secuencia.

2.2.1. Métodos basados en el análisis de la estructura

Los miRNA son procesados a partir de un precursor que tiene una estructura secundaria tipo tallo-horquilla que presenta ciertas particularidades. Por este motivo muchos métodos existentes utilizan las características de esta es-

tructura para identificar a los miRNA. Existen varios algoritmos que simulan el proceso de plegado: mínima energía libre [41], función de partición [22] y el algoritmo de plegado subóptimo [35], todos basados en programación dinámica.

Para extraer las características de este grupo se ha simulado el plegado mediante un software externo[1] y se cuenta con la estructura secundaria. Ésta se representa utilizando cadenas formadas por los símbolos '(', ')' y '.'. En estas cadenas, los '.' significan que un nucleótido no está emparejado y un par de paréntesis representan dos nucleótidos emparejados.

También existe otra sintaxis alternativa llamada sintaxis de Huang, propuesta en [15] que se utiliza en la extracción de varias características. En esta sintaxis se traduce la notación de paréntesis a una que utiliza los símbolos: '=' (emparejado), ':' (no emparejado), '.'(insertado), '-'(eliminado) y '^'(bucle) para marcar los estados de cada nucleótido en la estructura secundaria predicha. Por ejemplo, ante un tallo perfecto sólo tendremos símbolos '=', si aparece un bulto tendremos '.' del lado del bulto y '-' en el otro brazo. En el caso de bucles internos simétricos tendremos ':' por cada par de bases no emparejadas. Si los bucles son asimétricos del lado que haya más bases sin emparejar se completa con '.' y el otro brazo con '^'. Al finalizar, ambos brazos deben tener la misma cantidad de símbolos.

A continuación se presenta la lista de características basadas en el análisis de la secuencia primaria y su estructura secundaria. Junto al nombre de cada característica se coloca entre paréntesis el símbolo o nombre abreviado con el que se lo encuentra en la bibliografía.

- **Proporción de nucleótidos (A %, C %, G %, U %)**: porcentaje de cada nucleótido en la secuencia con respecto a la cantidad total. Esta característica ha sido propuesta en [29].
- **Proporción de dinucleótidos (AA %, AU %, . . . , GG %)**: expresa el porcentaje de dinucleótidos de cada tipo en el tallo con respecto a la cantidad total. Un dinucleótido es simplemente un par de nucleótidos que forman una subcadena de la cadena que se está procesando. Existen 16 posibles combinaciones, dado los 4 tipos de bases. Esta característica ha sido utilizada en [28] y [36].
- **Contenido de G+C ($G + C_{content}$)**: expresa el porcentaje de guanina más citosina presente en la secuencia. Se calcula como

$$G + C_{content} = \frac{G + C}{G + C + A + U}$$

[1] Vienna RNA (http://www.tbi.univie.ac.at/RNA/)

donde cada letra representa la cantidad de cada base contenida en la secuencia. Esta característica ha sido utilizada en [13], [28], [11] y [36].

- **Proporción de G+C ($G + C_{ratio}$)**: expresa la proporción de guanina y citosina en comparación con la de adenina y uracilo. Se calcula como

$$G + C_{ratio} = \frac{A + U}{G + C}$$

donde cada letra representa la cantidad de cada base contenida en la secuencia. Esta característica ha sido propuesta en [16].

- **Longitud de la estructura secundaria (l)**: cantidad de nucleótidos en la estructura secundaria predicha. Esta característica ha sido propuesta en [11].

- **Longitud del tallo (l_s)**: cantidad de nucleótidos en la región del tallo de una estructura secundaria. Esta característica ha sido utilizada en [13], [31] y [11].

- **Promedio de pares base por tallo (Avg_BP_Stem)**: cantidad promedio de nucleótidos por tallo (no confundir con la región del tallo). Los tallos son regiones de una longitud mínima de 3 pares base donde todos los nucleótidos contiguos están emparejados. Esta característica ha sido utilizada en [29], [28], [16] y [36].

- **Longitud del tallo más largo**: cantidad máxima de nucleótidos consecutivos en la región del tallo de un pre-miRNA donde el emparejamiento sea perfecto. El proceso de extracción de esta característica consiste en buscar la mayor región del tallo libre de nucleótidos no emparejados y medir la longitud de esta. Esta característica ha sido propuesta en [29].

- **Longitud del bucle terminal (l_h)**: cantidad de nucleótidos no emparejados en la región del bucle terminal de una estructura secundaria. En función de las características anteriores

$$l_h = l - l_s.$$

Esta característica ha sido utilizada en [13], [31] y [11].

- **Número de bultos (N_b)**: cantidad total de bultos. Esta característica ha sido propuesta en [38].

- **Número de bucles (N_l)**: cantidad total de bucles, incluyendo el bucle terminal. Esta característica ha sido utilizada en [38] y [11].
- **Longitud del bucle más largo (l_{ll})**: es la longitud del bucle más largo en la estructura secundaria. El proceso de extracción consiste en buscar la porción de bases no emparejadas consecutivas más larga de la estructura secundaria y contar la cantidad de nucleótidos que la forman. Esta característica ha sido propuesta en [11].
- **Número de bucles asimétricos (N_{al})**: cantidad total de bucles asimétricos. Esta característica ha sido propuesta en [38].
- **Número de bucles simétricos (N_{sl})**: cantidad total de bucles simétricos. Esta característica ha sido propuesta en [11].
- **Número de nucleótidos en bucles simétricos (N_{nsl})**: cantidad de nucleótidos en bucles simétricos. Esta característica ha sido utilizada en [29] y [11].
- **Número de nucleótidos en bucles asimétricos (N_{nal})**: cantidad de nucleótidos en bucles asimétricos. Esta característica ha sido propuesta en [29].
- **Longitud de la región simétrica más larga**: longitud de la región más larga sin bucles asimétricos o bultos. Los bucles simétricos son permitidos. Esta característica ha sido propuesta en [29].
- **Distancia de la región simétrica más larga al bucle de la horquilla**: cantidad de pares desde la región sin bucles asimétricos más larga hasta el bucle de la horquilla. Esta característica ha sido propuesta en [29].
- **Tamaño promedio de bucles simétricos**: longitud promedio de los bucles simétricos de la estructura secundaria. El proceso de extracción consiste en contar la cantidad de nucleótidos no emparejados formando parte de bucles simétricos y dividir por la cantidad de bucles simétricos. Es decir, N_{nsl}/N_{sl}. Esta característica ha sido propuesta en [29].
- **Tamaño promedio de bucles asimétricos**: longitud promedio de los bucles asimétricos. El proceso de extracción consiste en contar las cantidades de nucleótidos no emparejados formando parte de bucles asimétricos y dividir por la cantidad de bucles asimétricos. Es decir, N_{nal}/N_{al}. Esta característica ha sido propuesta en [29].

- **Número de bultos de longitud 1 a 7 y >7**: es un vector de 8 elementos indicando la cantidad de bultos de longitud 1, 2, ..., 7 y mayor que 7. El proceso de extracción consiste en buscar todos los bultos en la estructura secundaria y para cada uno contar la cantidad de nucleótidos, para luego contar la cantidad absoluta de bucles de cada longitud. Esta característica ha sido propuesta en [38].

- **Número de bucles de longitud 1 a 7 y >7** : es un vector de 8 elementos indicando la cantidad de bucles de longitud 1, 2, ..., 7 y mayor que 7. El proceso de extracción consiste en buscar todos los bucles en la estructura secundaria y para cada uno contar la cantidad de nucleótidos, para luego contar la cantidad absoluta de bucles de cada longitud. Esta característica ha sido propuesta en [38].

- **Proporción de elementos de la sintaxis de Huang**: estas características almacenan información sobre la estructura del precursor. Se reemplaza la notación de paréntesis por la propuesta en [15] y luego se cuentan las ocurrencias de pares de símbolos. Este proceso se lleva a cabo sobre el tallo, el bucle terminal no se tiene en cuenta. Para extraer estas características se traduce la notación de paréntesis a la notación de Huang antes descripta. Luego se cuenta la cantidad de elementos distintos y se normalizan estos valores. Por elementos, en este caso se entiende pares consecutivos de símbolos. Existen 10 posibles elementos: "=-", "==", "=:", "_", "-=", "^^", "^=", ":.", ":^" y ":=" por lo que el resultado final de este método de extracción es un vector de 10 elementos. Esta característica ha sido propuesta en [15].

- **Porcentaje de emparejamiento de Huang (pMatch)**: esta característica utiliza la notación de Huang. Indica el emparejamiento de bases en un candidato a miRNA de 22 nucleótidos. Este candidato se elige como el que mayor nivel de emparejamiento tenga dentro de la estructura secundaria. El proceso de extracción consiste en mover una ventana de 22 nucleótidos sobre la estructura y en cada una se cuenta la cantidad de '='. El mayor porcentaje se toma como descriptor. Esta característica ha sido propuesta en [15].

- **Porcentaje de no emparejamiento de Huang (pMismatch)**: esta característica utiliza la notación de Huang. Indica el nivel de no emparejamiento de bases en un candidato a miRNA de 21 nucleótidos. Este candidato se elige como el que mayor nivel de no emparejamiento tenga dentro de la estructura secundaria. El proceso de extracción consiste en mover una ventana de 21 nucleótidos sobre la estructura y

en cada una se cuenta la cantidad de ':'. El mayor porcentaje se toma como descriptor. Esta característica ha sido propuesta en [15].

- **Porcentaje de eliminación/inserción de Huang (pDI)**: esta característica utiliza la notación de Huang. Indica la cantidad de eliminaciones/inserciones que hay en un candidato a miRNA de 21 nucleótidos. Este candidato se elige como el que mayor nivel de eliminaciones/inserciones tenga dentro de la estructura secundaria. El proceso de extracción consiste en mover una ventana de 21 nucleótidos sobre la estructura y en cada una se cuenta la cantidad de '.' o '-' que no tengan '^' en el brazo opuesto. El mayor número se toma como descriptor. Esta característica ha sido propuesta en [15].

- **Porcentaje de bultos de Huang (pBulge)**: indica la simetría de los bucles y bultos en un candidato a miRNA de 21 nucleótidos. Este candidato se elige como el más asimétrico. El proceso de extracción consiste en mover una ventana de 21 nucleótidos sobre la estructura y en cada una se cuenta la cantidad de '^' sobre el brazo izquierdo correspondiente a esta ventana. El mayor número se toma como descriptor. Esta característica ha sido propuesta en [15].

- **Número de pares base (nP)**: cantidad de pares de nucleótidos. El proceso de extracción consiste simplemente en contar la cantidad de nucleótidos emparejados y dividir este número por dos. Esta característica ha sido propuesta en [38].

- **Propensión de pares base ajustado (dP)**: mide la cantidad de pares de bases presentes en una estructura secundaria dividido la longitud de ésta en nucleótidos. Esto elimina el problema de que cadenas más largas suelen tener más bases emparejadas. Esta característica esta en el intervalo [0 – 0,5], siendo 0 el caso de que no haya ningún emparejamiento y 0,5 el caso donde se emparejen los l nucleótidos. Esta característica ha sido utilizada en [24], [28], [16] y [36].

- **Proporción de pares base ($A-U$, $C-G$, $G-U$)**: expresa el porcentaje de pares de cada tipo con respecto a la cantidad total. Notar la diferencia con la proporción de dinucleótidos, donde se mide la proporción de elementos conformados por dos bases consecutivas (en un mismo brazo de la estructura secundaria). Esta característica ha sido utilizada en [29], [28], [16] y [36].

- **Promedio de pares base por tallo ($A-U/N_{stems}$, $C-G/N_{stems}$, $G-U/N_{stems}$)**: expresa la cantidad de pares de cada tipo con respecto

a la cantidad de tallos en la estructura secundaria. Por tallo hablamos de regiones de longitud mayor o igual a 3 nucleótidos donde el emparejamiento es perfecto (no confundir con región del tallo). Esta característica ha sido utilizada en [29], [28], [16] y [36].

- **Contenido de $G + C$ en el bucle terminal**: mide el contenido de $G + C$ en la región del bucle terminal. El proceso es idéntico al de contenido de $G + C$ antes descrito, pero se aplica solo sobre la región del bucle terminal.Esta característica ha sido propuesta en [11].

- **Tripletas**: estas características consisten en contar la cantidad de apariciones de cada una de las 32 posibles tripletas y luego normalizar estas cantidades por la cantidad total de nucleótidos. Cada tripleta está formada por una de las 4 posibles bases seguida por la estructura de la tripleta. Esta última se construye con dos posibles símbolos, match (cualquiera de los dos paréntesis) o mismatch (un punto). El segundo corresponde al nucleótido central, el primero al anterior, y el tercero al posterior. Ejemplos de tripletas son "A((" o "U.(.". Esta característica ha sido utilizada en [5], [25], [11] y [16].

2.2.2. Métodos basados en el comportamiento termodinámico

Las características de este grupo se basan en la estabilidad termodinámica de la estructura secundaria obtenida al plegar la secuencia. En [39] se demuestra que las secuencias que contienen miRNA son más estables que otras secuencias. La mayoría de estas características requieren de un software externo para plegar la molécula de RNA.

- **Mínima energía libre de plegado (MFE)**: es la mínima energía libre alcanzada como resultado de predecir la estructura secundaria. Para calcularla existen varios métodos, entre ellos: mínima energía libre [41], función de partición [22] y el algoritmo de plegado subóptimo [35]. Esta característica ha sido utilizada en [29], [25], [15] y [11].

- **Energía libre del ensamble (EFE)**: es un resultado secundario obtenido al calcular la función de partición al momento del plegado. Una descripción detallada del proceso se puede ver en [22]. Esta característica ha sido utilizada en [28], [16] y [36].

- **Frecuencia de la estructura MFE en el conjunto (Freq)**: es otro resultado secundario obtenido al calcular la función de partición

al momento del plegado. Como antes se indicó, la descripción detallada se puede ver en [22]. Esta característica ha sido utilizada en [28], [16] y [36].

- **Diversidad del conjunto (Diversity)**: es otro resultado secundario del cálculo de la función de partición al momento del plegado. También se describe en [22]. Esta característica ha sido utilizada en [28], [16] y [36].

- **Diferencia entre MFE y EFE ($Diff$)**: es una simple diferencia entre estas dos características, dividido por la longitud de la secuencia,

$$Diff = \frac{MFE - EFE}{l}.$$

  Esta característica ha sido utilizada en [28], [16] y [36].

- **MFE ajustado (dG)**: es el cociente entre la MFE y la longitud de la cadena. Esto elimina el problema de que cadenas más largas suelen tener MFE menores. Se puede utilizar normalizada en [0, 1] o en [0, 100]. Se calcula como

$$dG = 100\frac{MFE}{l}.$$

  Esta característica ha sido utilizada en [10], [39], [24], [16] y [36].

- **Índice** $MFEI_1$: este índice se calcula como el cociente entre el MFE ajustado y el contenido de G+C. Esta característica ha sido utilizada en [39], [24], [28], [16] y [36].

- **Índice** $MFEI_2$: es un índice que proporciona información sobre la cantidad de energía libre que aporta cada tallo (N_{stems}). Se calcula como el cociente entre la MFE ajustada y la cantidad de tallos en la estructura secundaria. Esta característica ha sido utilizada en [24], [28], [16] y [36].

- **Índice** $MFEI_3$: es el cociente de la MFE ajustada y la cantidad de bucles

$$MFE_3 = \frac{dG}{N_l}.$$

  Esta característica ha sido utilizada en [28], [16] y [36].

- **Índice** $MFEI_4$: es el cociente de la MFE ajustada y la cantidad de pares de bases

$$MFE_4 = \frac{dG}{nP}.$$

Esta característica ha sido utilizada en [28], [16] y [36].

- **Entropía de Shannon ajustada (dQ)**: caracteriza la distribución de probabilidad de emparejamiento de bases en una estructura secundaria como un sistema caótico dinámico. Valores bajos de dQ corresponden a una distribución dominada por pocas bases con probabilidad de estar emparejadas. Estas bases son mejores predichas que aquellas que tienen múltiples estados alternativos. Se calcula como

$$dQ = \frac{1}{l} \sum_{i<j} p_{ij} \log_2 p_{ij},$$

donde p_{ij} representa la probabilidad de emparejamiento de las bases i y j, estimadas con el algoritmo de [22]. Esta característica ha sido utilizada en [24], [28] y [36].

- **Potencial de tallo izquierdo (P^L)**: es la máxima probabilidad de emparejamiento entre un nucleótido y cualquier otro que esté en el brazo 3'. El resultado es un vector donde el elemento i corresponde al resultado obtenido para el nucleótido ubicado en la posición i. Para calcular este valor, una vez plegada la secuencia con el algoritmo de función de partición [22], se busca para cada i la máxima probabilidad p_{ij} donde $j > i$. Esta característica ha sido propuesta en [10].

- **Potencial de tallo derecho (P^R)**: es la máxima probabilidad de emparejamiento entre un nucleótido y cualquier otro que esté en el brazo 5'. Al igual que en la característica anterior, el resultado es un vector donde el elemento i corresponde al resultado obtenido para el nucleótido ubicado en la posición i. La diferencia radica en que para calcular este valor, una vez plegada la secuencia con el algoritmo de función de partición [22], se busca para cada i la máxima probabilidad p_{ij} donde $j < i$. Esta característica ha sido propuesta en [10].

- **Potencial de bucle (V')**: representa el potencial que tiene una cierta posición de la secuencia de estar asociada con el bucle terminal de una estructura de horquilla. Como en las dos características anteriores, el resultado es un vector de la misma longitud que la secuencia. Para calcular el valor del elemento i del vector, una vez plegada la secuencia con el algoritmo de función de partición [22], se suman las probabilidades de emparejamiento de bases entre los dos lados de una región simétrica centrada en la posición i. Esta característica ha sido propuesta en [10].

2.2.3. Métodos basados en análisis estadísticos

Las características que siguen a continuación proporcionan información sobre la estabilidad de una estructura secundaria en comparación con otras estructuras aleatorias. El proceso es similar en todas las características, primero se toma algún indicador de la estabilidad termodinámica de la estructura que se esta analizando, luego se generan otras secuencias aleatorias basándose en la primera y se compara la estabilidad con la primera.

Para generar las secuencias aleatorias existen varios algoritmos. Algunos de éstos conservan la cantidad de bases de la cadena, otros la cantidad de dinucleótidos, otros no conservan estos valores pero los aproximan. Es importante tener en cuenta esto, dado que los precursores de miRNA deberían tener MFEs más bajas que las de secuencias aleatorias que conservan mononucleótidos y dinucleótidos. En cambio, en otros tipos de RNA no sucede esto. Por ejemplo, el mRNA tiene una MFE menor si conservamos mononucleótidos, pero no hay diferencias significativas si además se conservan dinucleótidos [34, 4]. Los algoritmos de generación de secuencias aleatorias que se implementaron en la biblioteca son:

- Barajado de mononucleótidos [34]: simplemente se permutan las posiciones de los nucleótidos de las bases repetidas veces. Este método conserva la proporción de bases, pero no la de dinucleótidos.
- Barajado de dinucleótidos [7]: este algoritmo preserva la composición exacta de mononucleótidos y dinucleótidos. Para realizar este barajado se itera sobre todos los posibles pares de nucleótidos (AA, AC, AG, etc.) y todos los trinucleótidos no solapados que comiencen y terminen con este par se permutan aleatoriamente. De esta forma los cambios no alteran la proporción de dinucleótidos de la secuencia.

Las características de este grupo son:

- **Puntaje de segmento o MFE normalizada (zMFE)**: calcula el z-score de la MFE. Este puntaje se define como la cantidad de desviaciones estándar que hay de diferencia entre la MFE de la secuencia analizada y la media de la distribución de MFEs de las secuencias aleatorias. El proceso de extracción consiste en generar un número grande de secuencias aleatorias (con alguno de los algoritmos que se explicaron antes) y calcular la MFE de cada una. Luego se calcula la media y la desviación estándar de esta distribución de valores y se combinan con la MFE de la secuencia que se quiere analizar

$$zMFE = \frac{MFE - \mu}{\sigma},$$

donde μ es la media de la distribución y σ la desviación estándar. Esta característica ha sido utilizada en [13] y [10].

- **EFE normalizada (zEFE)**: calcula el z-score de la EFE con el mismo proceso que en la característica anterior. Esta característica ha sido propuesta en [16]

- **MFE ajustada normalizada (zG)**: consiste en el z-score de la MFE ajustada con el mismo proceso que en la característica anterior. Esta característica ha sido utilizada en [24], [28] y [36].

- **Entropía de Shannon ajustada normalizada (zQ)**: consiste en el z-score de la Entropía de Shannon ajustada con el mismo proceso que en la característica anterior. Esta característica ha sido utilizada en [24], [28] y [36].

- **Propensión de pares base normalizada (zP)**: consiste en el z-score de la propensión de pares de bases ajustada con el mismo proceso que en la característica anterior. Esta característica ha sido utilizada en [24], [28] y [36].

- **Monte Carlo y test de aleatorización sobre MFE (pMFE)**: es una alternativa al z-score y, al igual que este, sirve para determinar si el valor de MFE de una secuencia es significativamente menor que el de una secuencia aleatoria. Se calcula como

$$pMFE = \frac{R}{N+1},$$

donde N es el número de secuencias aleatorias generadas y R el número de éstas que tiene una MFE menor o igual que la de la secuencia original. Esta característica ha sido propuesta en [4].

- **Monte Carlo y test de aleatorización sobre EFE (pEFE)**: aplica el mismo proceso que la característica anterior, pero sobre el EFE de una estructura secundaria. Esta característica ha sido propuesta en [16].

2.2.4. Métodos basados en conservación filogenética

Cuando dos secuencias de RNA de distintas especies presentan similitudes es posible que haya una relación funcional entre ellas. Por este motivo la conservación de una secuencia en el RNA de distintas especies es un buen indicador de que esa porción cumple una función importante, por ejemplo, es un miRNA.

Para buscar similitudes entre dos cadenas primero es necesario alinearlas. Se utiliza programación dinámica para encontrar la alineación óptima de las dos cadenas. Además se debe contar con una tabla de puntajes de emparejamiento y una penalización por huecos. Generalmente se le da un puntaje positivo a las coincidencias, uno nulo a las no coincidencias y un puntaje negativo a los huecos.

La extensión de estos algoritmos dinámicos a más de dos cadenas se limita a 8 o 10 secuencias, ya que para un número mayor el problema se vuelve computacionalmente intratable. Sin embargo, para estos casos se utilizan algoritmos heurísticos que van generando alineaciones progresivas hasta terminar de agregar todas las secuencias. Las características que siguen a continuación se extraen de grupos de secuencias ya alineadas.

- **Frecuencia de mutación**: esta característica sólo se puede utilizar en el caso de que se alinee estrictamente un par de secuencias. Mide la cantidad de mutaciones (diferencias) entre dos secuencias de RNA. Simplemente es el cociente entre el número de diferencias y la longitud de la cadena. Esta característica ha sido propuesta en [15].

- **Puntaje de conservación (CS)**: mide el nivel de conservación de nucleótidos entre varias secuencias alineadas de distintas especies. Para esto utiliza modelos ocultos de Markov filogenéticos. Son modelos espacio temporales, ya que describen la secuencia de RNA en dos procesos de Markov, uno que opera en la dimensión del tiempo (a lo largo de las ramas de un árbol evolutivo) y otro que opera en el espacio (a lo largo de la secuencia). El proceso completo se detalla en [30] y [40]. Esta característica ha sido propuesta en [10].

- **Entropías por columna de los brazos 5', 3' y región de horquilla (S5', S3', S0)**: es la entropía de cada brazo de la estructura secundaria y de la región de la horquilla calculada verticalmente entre las distintas secuencias que forman parte de la alineación. Primero es necesario plegar las secuencias para delimitar las distintas regiones. Luego se calcula la entropía de cada región mediante

$$S_\xi = \frac{-1}{N_\xi} \sum_{i \in \xi} \sum_{\alpha \in [ACGU]} p_{i\alpha} \ln p_{i\alpha},$$

  donde $p_{i\alpha}$ es la fracción del nucleótido α en la posición i de todas las secuencias alineadas, ξ es la región de interés y N_ξ es la longitud de esta región. Esta característica ha sido propuesta en [13].

- **Entropía mínima (S_{min})**: es la menor entropía de todas las subcadenas de 23 nucleótidos de una alineación. Para esta característica se busca en los brazos de la estructura de horquilla, la subcadena que contenga la mínima entropía. Se toma directamente esta entropía mínima como descriptor. Para calcular la entropía se utiliza la misma fórmula que en la característica anterior. Como el miRNA maduro se conserva muy bien entre distintas especias, una baja entropía indica mayor probabilidad de que la subcadena sea miRNA. Esta característica ha sido propuesta en [13].

- **Diferencia de estructura secundaria de dos secuencias alineadas ($Vstrc$)**: esta característica sólo se puede utilizar en el caso de que se alineen estrictamente 2 secuencias. Básicamente mide la diferencia entre sus estructuras secundarias. Para calcularla se obtienen las estructuras secundarias y se cuenta la cantidad de diferencias. Se realiza lo mismo sobre la secuencia primaria y luego se calcula el cociente

$$Vstrc = \frac{STR_{diff}}{SEQ_{diff}},$$

  donde STR_{diff} es la cantidad de diferencias en la estructura secundaria (en notación de paréntesis) y SEQ_{diff} es la cantidad de diferencias en las secuencias. Esta característica ha sido propuesta en [15].

- **Promedio de energía libre de plegado ($\bar{E}$)**: es el promedio de las MFE de varias secuencias alineadas. Esta característica ha sido propuesta en [13].

- **Diferencia de MFE de dos secuencias (VMFE)**: esta característica sólo se puede utilizar en el caso de que se alinee estrictamente un par de secuencias. Mide la diferencia de energía libre promedio por cada mutación. Se calcula como

$$VMFE = \frac{|MFE_a - MFE_b|}{SEQ_{diff}},$$

  donde MFE_a y MFE_b son las mínimas energías libres de cada secuencia y SEQ_{diff} es la cantidad de bases que sufrieron una mutación entre las dos secuencias. Esta característica ha sido propuesta en [15].

- **Promedio de los MFE ajustados ($\bar{\epsilon}$)**: es el promedio de las MFE ajustados de varias secuencias alineadas. Esta característica ha sido propuesta en [13].

- **Promedio del índice de $MFEI_1$ ($\bar{\eta}$)**: es el promedio de los $MFEI_1$ de varias secuencias alineadas. Esta característica ha sido propuesta en [13].
- **Energía libre de la estructura secundaria consensuada (E_{cons})**: mide la energía libre de la estructura secundaria consensuada entre varias secuencias alineadas. Para calcularla primero se deben alinear las secuencias para obtener la secuencia consensuada, luego esta se pliega para calcular la MFE. Esta característica ha sido utilizada en [13] y [21].

2.2.5. Métodos basados en candidatos a miRNA

Este grupo de característica se extrae de un posible candidato a miRNA maduro. En el trabajo presentado por [21], para obtener este candidato, se mueve una ventana de longitud l_w (aproximadamente 22 nucleótidos) sobre una estructura secundaria y se extraen sus características. Como resultado obtenemos un vector de longitud $l - l_w$ donde el elemento i representa el valor de la característica calculada en el candidato que comienza en la posición i de la secuencia y termina en la posición $i + l_w$.

- **Emparejamiento de bases**: es la suma de las probabilidades de emparejamiento de las bases que forman parte del candidato a miRNA maduro. Estas probabilidades se obtienen de la matriz de probabilidades de emparejamiento, resultado del algoritmo de [22]. Esta característica ha sido propuesta en [21].
- **No emparejamiento en el candidato**: cantidad de bases no emparejadas en el candidato a miRNA. Esta característica ha sido propuesta en [11].
- **Extensión del emparejamiento de bases**: es la suma de las probabilidades de emparejamiento de las bases que no forman parte del candidato a miRNA maduro, pero forman parte de la misma estructura secundaria. Estas probabilidades se obtienen también de la matriz de probabilidades de emparejamiento, resultado del algoritmo de [22]. Esta característica ha sido propuesta en [21].
- **Simetría de bultos**: esta característica da una idea de la simetría de la zona del candidato a miRNA. Se calcula como la diferencia de bases no emparejadas en el candidato a miRNA y las bases no emparejadas en el segmento correspondiente al otro brazo de la estructura. Esta característica ha sido propuesta en [21].

- **Distancia al bucle terminal**: es la cantidad de pares base entre el bucle terminal y el extremo más cercano del candidato a miRNA. Esta característica ha sido utilizada en [21] y [31].

- **Conservación del brazo 5' de un candidato**: número de bases conservadas entre dos o más secuencias en el brazo 5' de un candidato a miRNA, es decir, sin contar las 11 primeras bases. Cabe aclarar que, al aplicarse sobre dos o mas secuencias, es necesario alinearlas como se describió en el grupo anterior antes de contar la cantidad de bases conservadas. Esta característica ha sido propuesta en [21].

- **Conservación del brazo 3' de un candidato**: número de bases conservadas entre dos o más secuencias en el brazo 3' de un candidato a miRNA, es decir, sin contar las últimas 11 bases. Al igual que en la característica anterior, es necesario realizar una alineación de secuencias antes de extraer esta característica. Esta característica ha sido propuesta en [21].

Capítulo 3

Desarrollo de la biblioteca

En este capítulo se presenta la biblioteca desarrollada y la interfaz gráfica de usuario. En primer lugar se explican cuáles fueron las tecnologías utilizadas en el desarrollo: el lenguaje de programación, bibliotecas, software externo utilizado para realizar cálculos complejos, etc. Luego se describe la estructura de la biblioteca y su utilización. Por último se presenta la interfaz gráfica desarrollada y su funcionamiento.

3.1. Análisis y diseño de la solución

En esta sección se describirán algunas cuestiones básicas relacionadas con el desarrollo de la biblioteca y la interfaz gráfica: las tecnologías utilizadas, otras bibliotecas y el software externo.

3.1.1. Tecnologías utilizadas

En primer lugar hay que aclarar que la biblioteca se programó en lenguaje Matlab como un conjunto de funciones. No se utilizó el paradigma orientado a objetos porque el lenguaje seleccionado tiene su fuerte en la programación estructurada. Se decidió utilizar este lenguaje por varios motivos. En primer lugar, por que es el lenguaje más utilizado en el SINC (lugar donde se desarrolló el proyecto). En segundo lugar, el software Matlab es un estándar de facto en aplicaciones de ingeniería, ya que permite con una interfaz muy amigable y sencilla, simular y depurar algoritmos de alta complejidad. Además cuenta con bibliotecas de algoritmos básicos de gran calidad y optimización, lo que permite un rápido desarrollo y prueba. Por último, se cuenta con experiencia en la utilización de este lenguaje.

Para desarrollar la interfaz gráfica de usuario se decidió utilizar el módulo

GUIDE que provee el mismo entorno de Matlab. Se tomó esta decisión, en primer lugar, para usar el mismo lenguaje que el de la biblioteca y de esta forma evitar generar una interfaz entre distintos lenguajes. En segundo lugar, Matlab tiene incorporado un sistema de graficación muy completo que simplifica mucho la tarea de presentar los resultados. Además Matlab cuenta con funciones para leer y escribir en formato *fasta* y separado por comas (*csv*, del inglés *Coma Separate Values*) que, como veremos más adelante, son los formatos elegidos para la entrada de secuencias y la salida de resultados. Para leer los archivos de configuración en formato YAML se utilizó la biblioteca YAMLMatlab[1] distribuida libremente bajo licencia MIT.

3.1.2. Software externo

Se utilizaron los siguientes programas externos para realizar algunos de los procesos necesarios en la extracción de características: , para luego contar la cantidad absoluta de bucles de cada longitud

- Vienna RNA[2] [14]: de este paquete de software se utilizaron los programas RNAfold para realizar la predicción de estructura secundaria de las secuencias y RNAalifold para realizar las mismas predicciones sobre grupos de secuencias alineadas. Estos programas implementan los algoritmos de mínima energía libre [41] y función de partición [22]. Se utilizó uno u otro dependiendo de la característica que se necesita extraer. Se eligió este software por ser el más utilizado en la actualidad. Además se distribuye libremente bajo licencia pública general GNU.

- Clustal Omega[3] [32]: este software se utilizó para realizar la alineación de secuencias. Al igual que el paquete Vienna RNA, se lo eligió por ser el más utilizado en el campo y se encuentra en constante desarrollo. Se distribuye libremente bajo licencia pública general GNU.

- PHAST[4]: de este paquete de software se utilizó el programa *phyloFit* para calcular los puntajes de conservación de las secuencias alineadas. Se encontraron varias alternativas para realizar este proceso, pero se eligió este programa por ser el único libre y gratuito. Se distribuye bajo licencia BSD.

La interfaz con estos programas se realizó creando archivos con los datos, llamando a los programas como procesos separados y luego leyendo los archivos

[1] https://code.google.com/p/yamlmatlab/
[2] http://www.tbi.univie.ac.at/RNA/
[3] http://www.clustal.org/omega/
[4] http://compgen.bscb.cornell.edu/phast/index.php

creados con los resultados. En un primer momento se intentó crear bibliotecas dinámicas a partir del código de los programas y enlazarlas a Matlab para evitar el costo que tiene la escritura y lectura de archivos, pero se descartó esta opción porque la complejidad del código fuente original imposibilitaba adaptarlo en tiempos razonables para el alcance de este Proyecto Final. Además, al utilizar estos programas como procesos separados de la biblioteca, ésta se vuelve independiente del sistema operativo y puede funcionar en cualquier máquina que tenga el software instalado. Por último, esta forma de comunicación con los programas externos permite actualizaciones automáticas (por ejemplo mediante repositorios PPA) sin mayores complicaciones.

3.1.3. Diseño de la biblioteca

Si analizamos las características desarrolladas en el capítulo anterior, podemos observar que hay tres tipos de análisis realizables: i) de secuencia ii) de candidato y iii) de alineación. Por este motivo las funciones de la biblioteca se dividen en tres grupos independientes, cada uno relacionado con un tipo de análisis distinto, pero que siguen una estructura similar. Desde el punto de vista del usuario, la diferencia más importante entre estos grupos está en los datos de entrada.

El primero de estos grupos, está relacionado con el análisis de porciones de RNA y toma como entrada simplemente una secuencia de nucleótidos. Ésta debe tener una longitud aproximada de 70 nucleótidos y la estructura secundaria predicha debe ser del tipo tallo-bucle. La biblioteca no verifica estas cuestiones por lo que se podría utilizar con secuencias que no cumplan estos requerimientos, pero algunos procesos pueden dar resultados inesperados. Estas restricciones se deben a que los pre-miRNA generalmente tienen una estructura que presenta estas características [13].

El segundo grupo incluye todas las características extraídas de un grupo de secuencias alineadas. Al igual que el primer grupo, estas secuencias deben tener una longitud aproximada de 80 nucleótidos y ser plegables en una estructura tipo tallo-bucle.

El último grupo incluye las características que se pueden extraer de un candidato a miRNA, es decir, una porción del pre-miRNA de aproximadamente 23 nucleótidos de longitud. En el trabajo presentado por [21], se toma como longitud 23 y el índice de inicio se va moviendo sobre la secuencia obteniendo de esta forma todos los posibles candidatos.

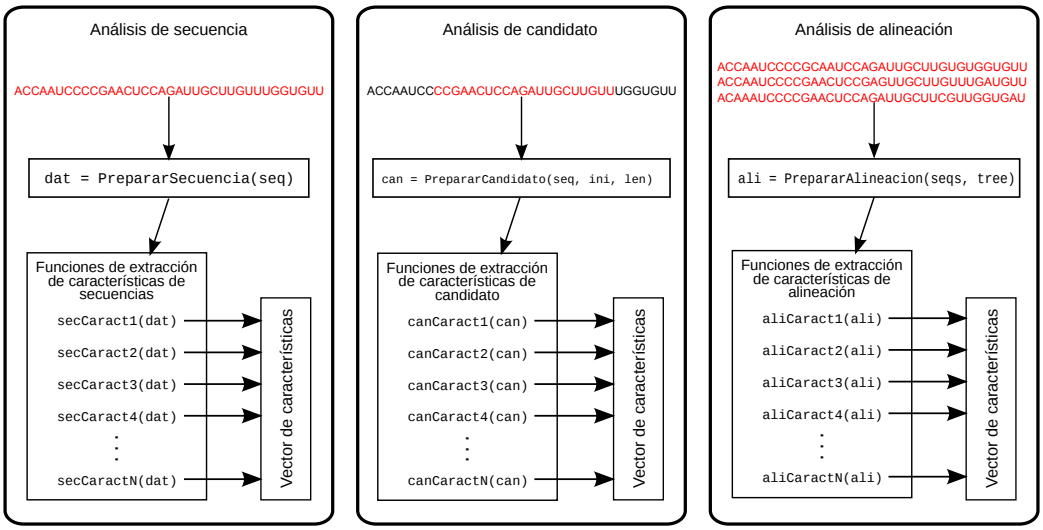


Figura 3.1: esquema de utilización de la biblioteca para realizar cada tipo de análisis

3.1.4. Estructura de la biblioteca

La biblioteca se compone de 71 funciones, de las cuales 5 son auxiliares y el resto se utiliza para extraer alguna característica. De estas 5 funciones auxiliares, 2 implementan los procesos de barajado aleatorio que precisan las funciones de análisis estadístico. Las otras 3 se encargan de preparar los datos de entrada y construir una estructura que es la que luego toman como parámetro de entrada el resto de las funciones. En la Figura 3.1 podemos ver la estructura general de la biblioteca y los procesos por los que debe pasar una secuencia (o grupo de secuencias) para extraer un vector de características. Dependiendo del tipo de análisis que queremos realizar, se debe utilizar una de las siguientes funciones:

- `function [dat] = PrepararSecuencia( seq )`: esta función toma como parámetro una secuencia y la pliega utilizando el algoritmo de función de partición implementado en el software *RNAfold*. De esta forma obtiene la estructura secundaria en notación de paréntesis, la matriz de probabilidad de emparejamiento y otras características termodinámicas de la secuencia (*MFE*, *EFE*, *Freq* y *Diversity*). Luego utilizando el Algoritmo 1 genera a partir de la estructura secundaria predicha, la misma en la notación alternativa de Huang. Todos estos resultados los almacena en el arreglo de celdas `dat` que devuelve como resultado.

- `function [ali] = AlinearSecuencias( seqs [, tree])`: esta fun-

Algoritmo 1: Huang

Datos: Estructura secundaria en notación de paréntesis
Resultado: Estructura secundaria en notación de Huang
inicio
- $b3 \leftarrow$ ‘’
- $b5 \leftarrow$ ‘’
- $i3 \leftarrow$ posición del primer paréntesis
- $i5 \leftarrow$ posición del último paréntesis
- **mientras** *i3 i5 no alcancen la region del bucle terminal* **hacer**
  - **si** *i3 = ‘(’ y i5 = ‘)’* **entonces**
    - b3 y b5 anexan ‘=’
    - $i3 \leftarrow$ siguiente nucleótido
    - $i5 \leftarrow$ nucleótido anterior
  - **sinó, si** *i3 = ‘.’ y i5 = ‘.’* **entonces**
    - b3 y b5 anexan ‘:’
    - $i3 \leftarrow$ siguiente nucleótido
    - $i5 \leftarrow$ nucleótido anterior
  - **sinó, si** *i3 = ‘(’ y i5 = ‘.’* **entonces**
    - **si** *Último agregado a b3 es ‘:’ o ‘^’* **entonces**
      - b3 anexa ‘^’ y b5 anexa ‘.’
    - **en otro caso**
      - b3 anexa ‘.’ y b5 anexa ‘-’
    - $i5 \leftarrow$ nucleótido anterior
  - **sinó, si** *i3 = ‘.’ y i5 = ‘)’* **entonces**
    - **si** *Último agregado a b5 es ‘:’ o ‘^’* **entonces**
      - b5 anexa ‘^’ y b3 anexa ‘.’
    - **en otro caso**
      - b5 anexa ‘.’ y b3 anexa ‘-’
    - $i3 \leftarrow$ siguiente nucleótido
- return $b3$ y $b5$

ción toma como parámetro un arreglo de secuencias. En primer lugar las alinea utilizando el software *Clustalω*. La secuencia consensuada resultado se pliega utilizando el software *RNAalifold*. Por último, si se especificó la topología del árbol filogenético en el parámetro `tree` (notar que es opcional), se calcula el puntaje de conservación con el software *phyloFit*. Un árbol filogenético es un árbol que muestra las relaciones evolutivas entre varias especies u otras entidades que se cree que tienen una ascendencia común. La topología debe estar representada por una cadena en formato Newick. Al igual que en la función anterior, los resultados se almacenan en un arreglo de celdas (`ali`).

- `function [can] = PrepararCandidato(dat, ini[, len])`: esta función toma como parámetro el arreglo de celdas resultado de la función `PraparарSecuencia`, el índice del nucleótido donde inicia el candidato que se quiere analizar y la longitud de este. Si no se especifica esta última se toma por omisión una longitud de 22 nucleótidos. No realiza ningún cálculo extra, sólo comprueba que los limites del candidato no caigan fuera de la secuencia y almacena los datos dentro del arreglo de celdas que devuelve como resultado (`can`).

El resto de las funciones toma estos arreglos y calcula las características solicitadas.

3.2. Funciones de extracción de características

3.2.1. Funciones de análisis de secuencias

Las funciones de este grupo extraen características de secuencias. Toman como parámetro el arreglo de celdas devuelto por la función `PrepararSecuencia`.

```
% Métodos basados en el análisis de la estructura
function [ pp ] = ProporcionNucleotidos( dat )
function [ r ] = ProporcionDinucleotidos( dat )
function [ r ] = ContenidoGC( dat )
function [ r ] = ProporcionGC( dat )
function [ r ] = Longitud( dat )
function [ r ] = LongitudTallo( dat )
function [ r ] = PromedioEmparejadosTallo( dat )
function [ r ] = LongitudTalloMasLargo( dat )
function [ r ] = LongitudBucleTerminal( dat )
function [ r ] = NumeroBultos( dat )
```

```
function [ r ] = NumeroBucles( dat )
function [ r ] = LongitudBucleMasLargo( dat )
function [ r ] = NumeroABucles( dat )
function [ r ] = NumeroSBucles( dat )
function [ r ] = NumeroBasesABucles( dat )
function [ r ] = NumeroBasesSBucles( dat )
function [mL, imL] = MaximaRegionSimetrica( dat )
function [ res ] = PromedioBasesABucles( dat )
function [ res ] = PromedioBasesSBucles( dat )
function [ nBucles ] = BuclesDeLongitud( dat )
function [ nBultos ] = BultosDeLongitud( dat )
function [ r ] = ProporcionElementosHuang( dat )
function [pMatch,pMismatch,pDI,pBulge]=ProporcionesHuang(dat)
function [ x ] = PotencialBucle( dat )
function [ x ] = PotencialTalloDerecho( dat )
function [ x ] = PotencialTalloIzquierdo( dat )
function [ nbp ] = NumeroEmparejados( dat )
function [ dP ] = PropensionParesBases( dat )
function [ ppb ] = ProporcionParesBases( dat )
function [ ppt ] = PromedioParesBasesTallo( dat )
function [ r ] = ContenidoGCBucleTerminal( dat )
function [ tr ] = Tripletas( dat )

% Métodos basados en el comportamiento termodinámico
function [ MFE ] = EnergiaLibrePlegado( dat )
function [ EFE ] = EnergiaLibreConjunto( dat )
function [ FREQ ] = FrecuenciaConjunto( dat )
function [ Diversity ] = DiversidadConjunto( dat )
function [ Diff ] = DiferenciaMFE_EFE( dat )
function [ dG ] = EnergiaLibrePlegadoAjustada( dat )
function [ MFE1 ] = IndiceMFE1( dat )
function [ MFE2 ] = IndiceMFE2( dat )
function [ MFE3 ] = IndiceMFE3( dat )
function [ MFE4 ] = IndiceMFE4( dat )
function [ dQ ] = EntropiaShannonAjustada( dat )
```

El siguiente grupo de funciones además de tomar el arreglo de celdas como parámetro, toman también la cantidad de iteraciones y el método de barajado.

```
function [ r ] = pvalueMFE( dat, iter, Shuffle )
```

```
function [ r ] = pvalueEFE( dat, iter, Shuffle )
function [ r ] = zEFE( dat, iter, Shuffle )
function [ r ] = zG( dat, iter, Shuffle )
function [ r ] = zMFE( dat, iter, Shuffle )
function [ r ] = zP( dat, iter, Shuffle )
function [ r ] = zQ( dat, iter, Shuffle )
```

Los métodos de barajado que implementa la biblioteca son:

```
function [ seq ] = DiShuffle( seq )
function [ seq ] = MonoShuffle( seq )
```

3.2.2. Funciones de análisis de secuencias alineadas

Las funciones de este grupo extraen características de grupos de secuencias alineadas. Toman como parámetro el arreglo de celdas devuelto por la función `PrepararAlineacion`.

```
% Métodos basados en conservación filogenética
function [ pMutFeq ] = FrecuenciaDeMutacion( ali )
function [ cs ] = PuntajeConservacion( ali )
function [ S5, S0, S3, Smin ] = EntropiaPorColumna( ali )
function [ Vstrc ] = DiferenciaDeEstructuraSecundaria( ali )
function [ Eprom ] = MFEPromedio( ali )
function [ VMFE ] = MFEDiferencia( ali )
function [ MFEap ] = MFEAjustadoPromedio( ali )
function [ MFE1p ] = IndiceMFE1Promedio( ali )
function [ MFE ] = EnergiaEstructuraConsensuada( ali )
```

3.2.3. Funciones de análisis de candidatos a miRNA

Las funciones de este grupo extraen características de candidatos a miRNA.
Toman como parámetro el arreglo de celdas devuelto por la función `PrepararCandidato`.

```
function [ r ] = CandidatoSimetriaBultos( can )
function [ r ] = CandidatoDistanciaBucle( can )
function [ r ] = CandidatoNoEmparejado( can )
function [ r ] = CandidatoEmparejamientoBases( can )
function [ r ] = CandidatoExtensionEmparejamiento( can )
function [ r ] = Conservacion3( ali )
function [ r ] = Conservacion5( ali )
```

Algoritmo 2: Algoritmo que calcula $zMFE$ en paralelo

Datos: *dat*: arreglo de celdas, *iter*: número de iteraciones y
barajar: función de barajado
Resultado: $zMFE$
inicio
  $MFE\_original \leftarrow MFE(dat)$
  $distribución \leftarrow$ vector nulo de longitud $iter$
  para $i \leftarrow 0$ **a** $iter$ **hacer en paralelo**
    $secuencia\_aleatoria \leftarrow barajar(dat)$
    $distribución_i \leftarrow MFE(secuencia\_aleatoria)$
  $zMFE \leftarrow \frac{MFE\_original - media(distribución)}{varianza(distribución)}$

3.3. Paralelización

Las funciones que realizan pruebas estadísticas generalmente se utilizan con un número de iteraciones grande para que los resultados sean significativos. Esto implica que estas funciones deben generar miles de secuencias aleatorias, plegarlas para extraer la característica de interés y luego analizar la distribución de estos valores. Estos cálculos precisan de un importante tiempo de computo. Además, la estructura de estas funciones es simple de dividir en múltiples trabajadores. Por estos motivos se decidió paralelizarlas utilizando las funciones de cálculo paralelo de Matlab.

Tanto en el caso de las pruebas de z-score como en las de Monte Carlo, se dividió el trabajo de generar las secuencias aleatorias y extraer las características de éstas en distintos hilos de ejecución. Se muestra para ejemplificar el pseudo código de los algoritmos que normalizan la mínima energía libre en los Algoritmos 2 y 3. La estructura es la misma para calcular $pEFE$, $zEFE$, zG, zP y zQ.

3.4. Interfaz de usuario final

En esta sección se presentará la interfaz gráfica de usuario. Ésta fue desarrollada teniendo presente que está dirigida a usuarios con pocos conocimientos de informática, de forma que puedan acceder fácil y rápidamente a las funciones de la biblioteca. Describiremos la utilización de los archivos de configuración, después como se resolvió la entrada y salida de datos y por último una descripción de la interfaz gráfica y las opciones que presenta.

Algoritmo 3: Algoritmo que calcula $pMFE$ en paralelo

Datos: *dat*: arreglo de celdas, *iter*: número de iteraciones y
barajar: función de barajado
Resultado: $pMFE$
inicio
  $MFE\_original \leftarrow MFE(dat)$
  $R \leftarrow 0$
  para $i \leftarrow 0$ **a** *iter* **hacer en paralelo**
    $secuencia\_aleatoria \leftarrow barajar(dat)$
    si $MFE(secuencia\_aleatoria) \leq MFE\_original$ **entonces**
      $R \leftarrow R + 1$
  $pMFE \leftarrow \frac{R}{iter+1}$

3.4.1. Archivos de configuraciones

Se decidió utilizar archivos de configuración en formato YAML para manejar la gran cantidad de opciones que presenta la biblioteca. Éste es un formato de texto plano de lectura humana directa, por lo que es fácil de modificar con cualquier editor de texto. De esta forma evitamos complejas interfaces gráficas y además nos permite guardar distintas configuraciones de pruebas, copiarlas y trasladarlas de una máquina a otra sin mayores complicaciones. En la Figura 3.2 se muestra un archivo de configuración modelo que se entrega junto con la biblioteca para usar de plantilla.

El archivo tiene una estructura jerárquica simple de leer, al mismo tiempo que permite manejar una gran cantidad de opciones. Dentro del mismo archivo hay líneas con comentarios (comienzan con el caracter numeral) que explican para que se utiliza cada opción.

3.4.2. Formatos de entrada y salida

Para manejar la entrada de datos se eligió el formato *fasta*. Este es un formato de archivo estándar basado en texto, utilizado para representar cadenas de nucleótidos o cadenas de aminoácidos, donde cada componente de la cadena se representa con un caracter. Este formato también permite agregar nombres a las secuencias y comentarios opcionales. Se tomó este formato porque es un estándar en el campo de la bioinformática, además de ser simple de leer desde cualquier programa. Matlab cuenta con una función especial para leer este tipo de archivos.

Para la salida de datos se eligió el formato *csv* por ser un formato sim-

```
---
  # Nombre del análisis.
  # Aparecerá en ventanas de informe y archivos de salida
  Nombre:  AlineaciónTodo
  # Tipo de análisis (``Secuencia'', ``Candidato'' o ``Alineación'').
  # El resto del formato del archivo depende de esta opción.
  Tipo: Alineación
  OpcionesPlegado:
        # Reescalar los parámetros de energía a una temperatura personalizada.
        PersonalizarTemperatura: no
        Temperatura: 37
        # No permitir pares G-U.
        NoGU: no
        # No permitir pares G-U al final de los brazos.
        NoCloseGU: no
  OpcionesAlineacion:
        # Usar matriz de distancia completa para los cálculos del árbol guía (lento)
        FullDistanceMatrix: no
        # Usar matriz de distancia completa para los cálculos
        # del árbol guía durante la iteración(lento)
        FullDistanceMatrixIter: no
        # Usar distancia de Kimura para la corrección de secuencias alineadas
        UseKimura: no
        # Numero de iteraciones (-1 para ignorar).
        MaxIterations: -1
        # Máximo número de iteraciones en el árbol guía (-1 para ignorar).
        MaxGTIterations: -1
        # Máximo número de iteraciones en el HMM (-1 para ignorar).
        MaxHMMIterations: -1
   # Características a extraer. Valores permitidos ``yes'' o ``no''.
  Caracteristicas:
        FrecuenciaDeMutacion: yes
        PuntajeConservacion:
            ejecutar: no
            # Arbol de ejemplo, ver PhyloFit --help para mas ayuda
            arbol: (human, (mouse,rat)mouse-rat, cow)
        EntropiaPorColumna: yes
        DiferenciaEstructuraSecundaria: yes
        MFEPromedio: yes
        MFEDiferencia: yes
        MFEAjustadoPromedio: yes
        IndiceMFE1Promedio: yes
        EnergiaEstructuraConsensuada: yes
        Conservacion3: yes
        Conservacion5: yes
  Posproceso:
        Normalizado: no
```

Figura 3.2: modelo de archivo de configuración de un análisis de alineación de secuencias

ple, de lectura humana directa y compatible con gran cantidad de software. Además Matlab cuenta con funciones especiales para escribir y leer en este formato. Por cada prueba se generan tres archivos en formato *csv*: i) un archivo con los resultados ii) un archivo con la etiqueta que indica a que característica corresponde cada resultado y iii) una lista de las características que se extrajeron en el análisis.

El programa coloca automáticamente el nombre a los archivos de salida, formándolo con el nombre de la prueba (tomado del archivo de configuración), el nombre del archivo de secuencias de entrada, una marca de tiempo y una palabra que indica a cual de los tres archivos corresponde. Esta última es "*result*" para el archivo que contiene los resultados, "*tags*" para el archivo que contiene las etiquetas y "*map*" para el archivo que tiene los nombres de las características que se extrajeron. Por ejemplo, si utilizamos el archivo de configuración de la Figura 3.2 y un archivo de secuencias de nombre "*Echi_small.fasta*", los archivos de salida tendrán como nombre "*AlineaciónTodo_Echi_small_2014-2-11-19193_map.csv*", "*AlineaciónTodo_Echi_small_2014-2-11-19193_result.csv*" y "*AlineaciónTodo_Echi_small_2014-2-11-19193_tags.csv*"

3.4.3. Descripción de la interfaz gráfica de usuario

La interfaz gráfica de usuario es bastante simple pero permite acceder a todas las funciones de la biblioteca y agrega además algunas características. Como se puede ver en la Figura 3.3, en a) presenta la opción de seleccionar un archivo de configuración y de secuencias de entrada (en los formatos descriptos en la sección anterior). Además de contar con dos cuadros de texto donde tenemos la posibilidad de escribir la dirección de los archivos, se agregaron dos botones cuyo icono es una carpeta dado que la amplia mayoría de los usuarios relacionará este símbolo con la acción de abrir. Con estos botones se muestra un dialogo de selección de archivos similar al del los sistemas operativos Microsoft Windows®. Cualquier usuario que haya utilizado este sistema operativo inmediatamente descubrirá como seleccionar el archivo deseado. Además este dialogo solo muestra los archivos que tengan el formato adecuado (*yaml* o *fasta*, dependiendo del caso) para evitar errores del usuario.

En b) presenta una lista de tareas y botones que permiten agregar nuevas, quitarlas o ejecutarlas. Esta lista puede cargar varios trabajos para ejecutarlos luego por lotes, lo que es útil cuando se desea analizar una gran cantidad de secuencias con distintos archivos de configuración. Los nombres de los botones que acompañan la lista son autoexplicativos, por lo que un usuario común no debería tener problemas para comprender su funcionamiento.

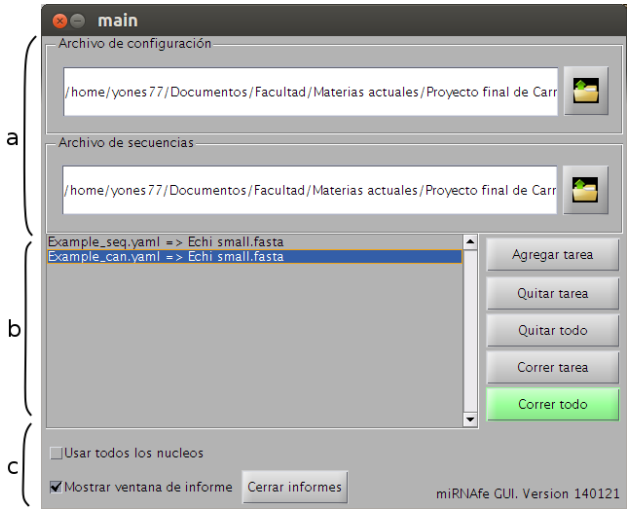


Figura 3.3: ventana principal de la aplicación; (a) campos de selección de archivos de entrada y configuración (b) lista de trabajos (c) opciones extra.

En c) encontramos dos opciones: la primera permite que algunos procesos de extracción de características se ejecuten en paralelo para reducir los tiempos de procesamiento en máquinas con múltiples núcleos; la segunda indica si se debe mostrar una ventana de informe al finalizar cada prueba.

Si el archivo de entrada tiene más de 10 secuencias y la opción de presentar ventanas de informe para cada análisis esta tildada, al iniciar las pruebas se muestra una ventana de advertencia al usuario. Ésta indica que se va a mostrar una gran cantidad de ventanas de informe y pide una confirmación.

Mientras se ejecutan las pruebas se muestra una ventana con una barra de progreso que indica que secuencia se está analizando. Al finalizar cada tarea, si la opción "*Mostrar ventana de informe*" esta tildada, por cada secuencia se nos presentarán dos ventanas con los resultados de la extracción de características. Las ventanas de informe son distintas dependiendo de qué tipo de análisis se realiza (de secuencia, de candidato o de alineación) y de qué características se seleccionan en el archivo de configuración.

Se tomo una vista adecuada para cada tipo de característica:

- Las características que tienen una dimensión mayor a 32 y sus elementos son dependientes de la posición en el vector (por ejemplo, el elemento i corresponde al nucleótido i) se presentan en gráficos de líneas.

- Las que tienen una dimensión de a lo sumo 32 elementos o sus elementos tienen valores independientes de la posición (por ejemplo la proporción de tripletas) se muestran con gráficos de barras.

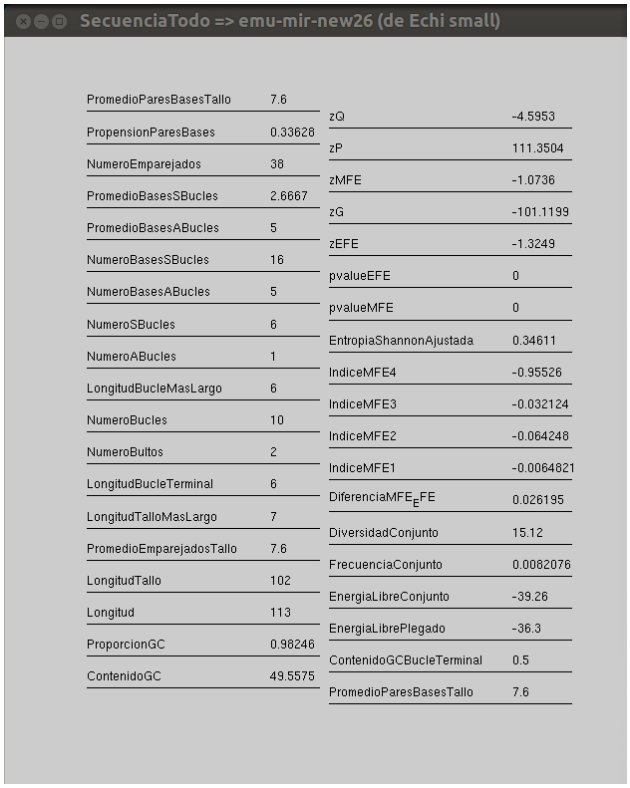


Figura 3.4: ventana de informe de características unidimensionales en un análisis de secuencia.

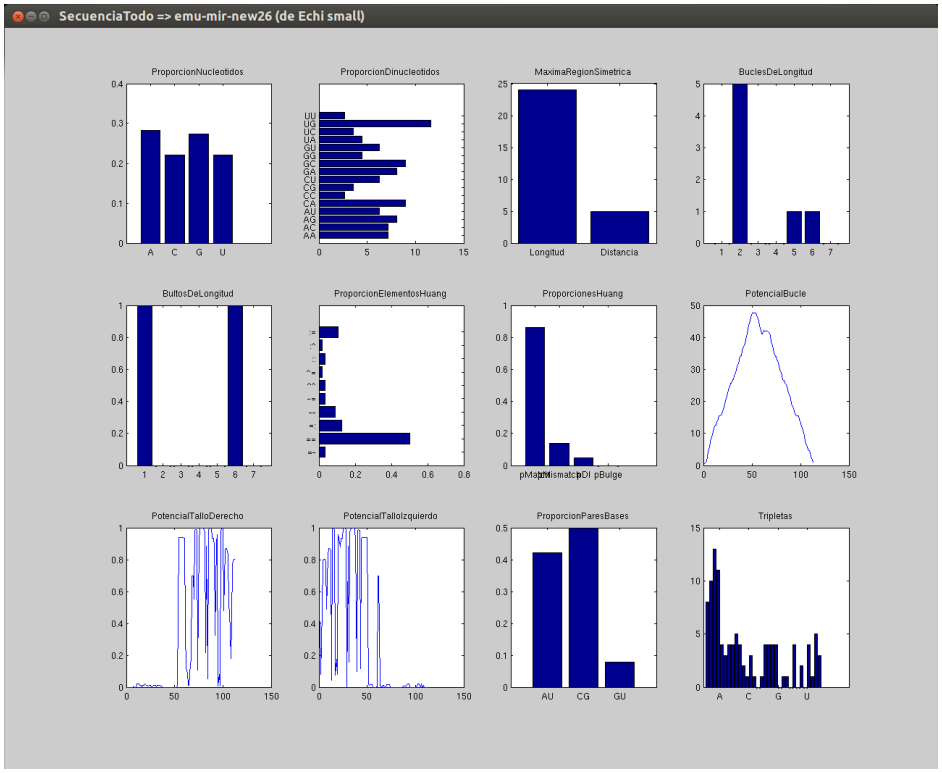


Figura 3.5: ventana de informe de características multidimensionales en un análisis de secuencia.

- Las unidimensionales se muestran en formato digital, en una ventana separada.

En las Figuras 3.4 y 3.5 se puede ver un ejemplo de las ventanas de informe para un análisis de secuencia donde se extrajeron todas las características disponibles para este tipo de análisis.

Capítulo 4

Resultados

En este capítulo se describe la metodología utilizada para realizar las pruebas y sus resultados. Además se presenta un resumen de los cambios y correcciones realizadas durante este proceso.

4.1. Metodología para la validación

Las pruebas se realizaron a nivel unitario utilizando el método de caja negra sobre las funciones de extracción de características. Se automatizo el proceso creando funciones en Matlab que se encargan de preparar las secuencias de entrada, ejecutar las pruebas y comparar con los resultados esperados. Además miden los tiempos de ejecución de cada función.

Vale aclarar que estas funciones de prueba cobran un papel importante no solo en el contexto de este trabajo, sino también a la hora de realizar mejoras o re-implementaciones de las funciones de la biblioteca ya que permiten verificar que los resultados sean correctos ante cada cambio que se haga.

Para cada prueba se tomó una tolerancia al error distinta, dependiendo de la naturaleza de los resultados. Por ejemplo, en el caso de las pruebas que cuentan cierto patrón dentro de la estructura secundaria, se pretende que el error sea cero, en cambio, en las funciones no determinísticas (como z-score y Monte Carlo) los resultados pueden variar de una ejecución a otra, por lo que se consideró algún nivel de tolerancia.

A medida que se va probando cada función, por consola se imprimen los resultados de ésta, los resultados esperados y el error. En la Figura 4.1 se puede ver la salida de las pruebas de algunas de las funciones.

Como medida del error se tomó el cociente entre la norma 2 del vector error y la norma 2 del vector con el valor esperado. Si este cociente es menor que 0,01 se considera que el resultado de la función es correcto. Para las

```
Probando IndiceMFE4...
Resultados obtenidos con los scripts genRNAStats y RNAspectral de miPred (Jiang 2007)
    Test número 1:
        Correcto:  -1.0364
        Calculado: -1.0364
        Error: 3.6364e-05 (%0 < %1)
                Resultado ==========> CORRECTO
    Test número 2:
        Correcto:  -1.1367
        Calculado: -1.1367
        Error: 3.3333e-05 (%0 < %1)
                Resultado ==========> CORRECTO
    Test número 3:
        Correcto:  -1.3735
        Calculado: -1.3735
        Error: 2.9412e-05 (%0 < %1)
                Resultado ==========> CORRECTO
Pruebas correctas: 3 / 3

Probando MFEDiferencia...
Resultados obtenidos con MiRFinder (Huang 2007)
    Test número 1:
        Correcto:  0
        Calculado: 0
        Error: 0 (%0 < %1)
                Resultado ==========> CORRECTO
    Test número 2:
        Correcto:  0.75
        Calculado: 0.75
        Error: 0 (%0 < %1)
                Resultado ==========> CORRECTO
    Test número 3:
        Correcto:  3.2
        Calculado: 3.2
        Error: 8.8818e-16 (%0 < %1)
                Resultado ==========> CORRECTO
Pruebas correctas: 3 / 3
```

Figura 4.1: salida por consola de las pruebas de dos funciones de la biblioteca

funciones no determinísticas esta tolerancia aumenta a 0,15. Si la norma del vector esperado es cero, la norma del resultado debe ser cero para considerarlo correcto. Vale aclarar que en la mayoría de las características, los vectores son unidimensionales.

Como entrada para las pruebas se seleccionaron 6 secuencias: ppa-mir-101, hsa-mir-34a, hsa-mir-7-1, hsa-let-7a-1, hsa-let-7a-3 y hsa-let-7b3. Se tomaron estas secuencias porque aparecen en la mayoría de las bases de datos de otros autores y de esta forma se cuenta con resultados fiables para comparar con los calculados con las funciones de la biblioteca. Además, es muy difícil saber que entradas pueden generar fallas y no se pueden crear clases de equivalencia dada la naturaleza de los datos, por lo que la elección de las secuencias no mejora el proceso de pruebas. Para cada algoritmo probado se tomaron 3 secuencias de las antes nombradas, eligiendolas de acuerdo a los resultados correctos con los que se contaba. Es decir, se elegian 3 secuencias de las 6 sobre las cuales se contaba con los resultados correctos para la característica probada. En los casos en los que existia mas de 3 posibilidades se elegian al azar.

Para obtener los resultados correctos se tomaron distintas estrategias de acuerdo al nivel de complejidad de los algoritmos involucrados. En el caso de las funciones más complejas, se compararon los resultados con los obtenidos por otros autores. Otro grupo de funciones (principalmente las que extraen características de la estructura secundaria) fueron comparadas con resultados obtenidos manualmente por el autor. Éste último grupo de funciones realiza procesos que pueden ser realizados por una persona, como contar la cantidad de bucles, la longitud de cierta región, la proporción de cada tipo de nucleótido, etc. Por último se listan algunas funciones que por distintos motivos no pudieron ser comparadas con otras referencias.

4.1.1. Comparación de resultados con otros autores

Para los algoritmos más complejos, como antes se indicó, se compararon las salidas con resultados obtenidos por otros autores. En algunos casos se consiguieron bases de datos con secuencias y sus características, en otros se tuvo que correr el software de los autores originales y comparar las salidas. A continuación se describen las fuentes de las características extraídas por otros autores y la lista de funciones que se probaron con éstas:

- Resultados obtenidos con el software MiRFinder [15]

  ```
  ProporcionElementosHuang
  ProporcionesHuang
  MFEDiferencia
  FrecuenciaDeMutacion
  DiferenciaDeEstructuraSecundaria
  ```

- Resultados obtenidos con el software miPred [25]

  ```
  Tripletas
  IndiceMFE1
  IndiceMFE2
  IndiceMFE4
  pvalueMFE
  pvalueEFE
  zEFE
  zG
  zMFE
  zP
  ```

```
zQ
```

- Resultados obtenidos con microPred [28]

```
ProporcionDinucleotidos
ContenidoGC
IndiceMFE3
```

4.1.2. Comparación de resultados obtenidos por inspección

Primero se plegaron las secuencias de prueba utilizando el software RNAfold con las opciones necesarias para que genere una imagen de la estructura secundaria. Luego se revisaron estas imágenes, realizando manualmente los procesos necesarios para obtener en cada caso los resultados esperados, que luego se compararán con los calculados por la biblioteca. Las funciones cuyos resultados fueron comparados son las siguientes:

```
ProporcionNucleotidos
LongitudTallo
PromedioEmparejadosTallo
LongitudTalloMasLargo
LongitudBucleTerminal
NumeroBultos
LongitudBucleMasLargo
NumeroABucles
NumeroSBucles
NumeroBasesABucles
NumeroBasesSBucles
MaximaRegionSimetrica
PromedioBasesABucles
PromedioBasesSBucles
BuclesDeLongitud
BultosDeLongitud
ProporcionParesBases
PromedioParesBasesTallo
ContenidoGCBucleTerminal
```

Para las funciones que analizan candidatos, se tomaron 5 porciones de la secuencia ppa-mir-101 y, al igual que en el caso anterior, se calcularon manualmente los resultados esperados. Las funciones probadas fueron:

```
CandidatoSimetriaBultos
CandidatoDistanciaBucle
CandidatoNoEmparejado
CandidatoEmparejamientoBases
CandidatoExtensionEmparejamiento
```

Por último, se probaron tres funciones que analizan alineaciones. Primero se alinearon y plegaron dos versiones (que presentaban pequeñas mutaciones) de las secuencias ppa-mir-101, hsa-mir-34a y hsa-mir-7-1. Estas se obtuvieron de la base de datos de secuencias de [15]. Luego se calcularon los valores esperados de las siguientes funciones:

```
Conservacion3
Conservacion5
EntropiaPorColumna
```

4.1.3. Funciones que no se prueban

En primer lugar, algunas funciones simplemente son una interfaz con otro software. Este tipo de funciones no necesita ser probada, ya que los resultados dependen exclusivamente de los programas externos utilizados y en los tres casos (RNAfold, ClustalW y PHAST), se trata de software avalado por trabajos científicos [14, 32, 40], ampliamente utilizados en el campo de la bioinformática. Luego tenemos otro grupo de funciones que aplican algoritmos sencillos (como diferencias, o promedios) a otras características para combinarlas. Como las características combinadas provienen de otras funciones que fueron probadas y al ser algoritmos codificados como una simple llamada a función de Matlab, se consideró que no es necesario realizar pruebas comparativas sobre estas funciones. Por último, algunas funciones son utilizadas por otras a las que sí se le realizaron pruebas, por lo que fueron probadas indirectamente. Como los resultados de las segundas son correctos, se puede suponer que los resultados de éstas también lo son.

A continuación se listan las funciones que están incluidas en cada grupo

- Funciones de interfaz con software externo:

  ```
  EnergiaLibreConjunto
  EnergiaLibrePlegado
  ```

```
FrecuenciaConjunto
DiversidadConjunto
PuntajeConservacion
EnergiaEstructuraConsensuada
PotencialBucle
PotencialTalloDerecho
PotencialTalloIzquierdo
EntropiaShannonAjustada
```

- Funciones que aplican algoritmos simples para combinar otras características:

```
DiferenciaMFE_EFE
MFEPromedio
MFEAjustadoPromedio
IndiceMFE1Promedio
PropensionParesBases
```

- Funciones probadas indirectamente (se usan en otras funciones cuyos resultados fueron comparados):

```
Longitud
MFEAjustada
NumeroBucles
NumeroEmparejados
NumeroTallos
```

4.2. Pruebas y análisis de resultados

Se encontraron diferencias entre los valores esperados y los resultados obtenidos en las siguientes funciones: BuclesDeLongitud, BultosDeLongitud, PrepararAlineación, NumeroBasesABucles, NumeroBasesSBucles, NúmeroTallos, NumeroBultos y NumeroBucles, ProporcionElementosHuang, ProporciónHuang, MFEDiferencia, DiferenciaEstructuraSecundaria y PrepararAlineación. En todos los casos se corrigieron los errores. Por otro lado, se reescribieron otras funciones para mejorar la legibilidad del código y/o la velocidad de ejecución: NumeroABucles, NumeroSBucles, MonoShuffle y DiShuffle.

Para aproximar los tiempos de ejecución de las funciones se promediaron los tiempos necesarios para procesar cada una de las tres secuencias de prueba. Los resultados obtenidos se presentan en la Figura 4.2. Como se observa en los gráficos, sólo nueve funciones tardan más de un milisegundo en procesar una secuencia. De éstas, dos (DiferenciaEstructuraSecundaria y FrecuenciaMutación) tardan aproximadamente 10 milisegundos. Las funciones que realizan análisis estadísticos son las que más tiempo requieren, superando los 4 segundos, y llegando a los 11 en el caso de zQ.

Como se pudo ver, las funciones que realizan pruebas estadísticas tenían tiempos de ejecución muy por encima del resto de las funciones. Tanto en el caso de las pruebas de z-score como en las de Monte Carlo, se dividió el trabajo de generar las secuencias aleatorias y extraer las características de éstas en distintos hilos de ejecución.

Se volvieron a medir los tiempos luego de modificar los algoritmos. Los tiempos de ejecución antes y después de la paralelización se pueden ver en la Figura 4.3 y la eficiencia, definida como la cantidad de hilos de ejecución (8) por el cociente entre el tiempo tardado despues de la paralelización sobre el tiempo tardado antes, en la Figura 4.4. Las pruebas se corrieron en una computadora con un microprocesador Intel® Core(TM) i7-3770 CPU @ 3.40GHz y 8 Gb de memoria RAM. Cabe aclarar que este procesador cuenta con 4 núcleos reales y 4 virtuales, resultado de la tecnología Intel® Hyper-Threading. Como se puede observar, en todos los casos la eficiencia es superior al 75 % y los tiempos de cómputo se han podido reducir significativamente.

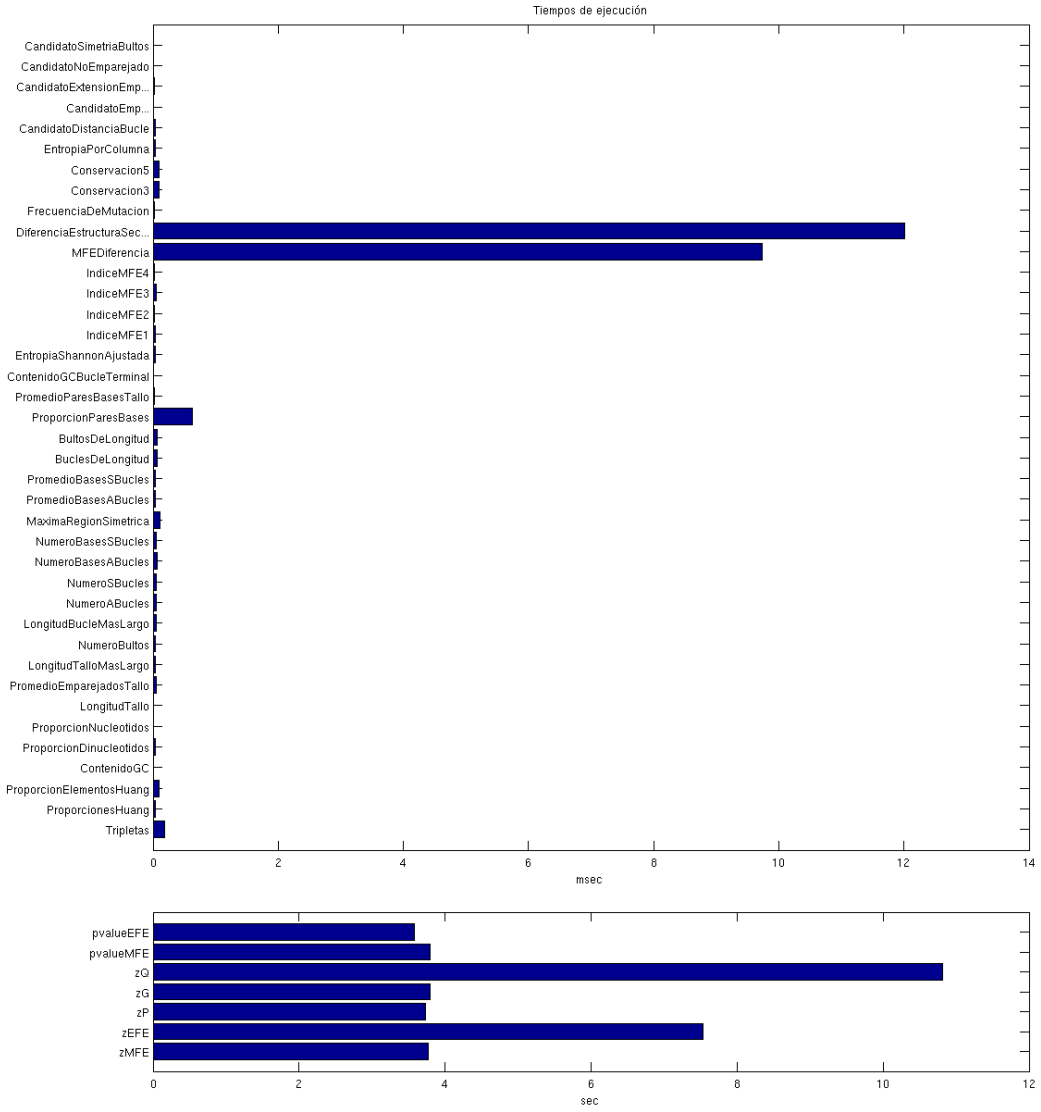


Figura 4.2: tiempos de ejecución.

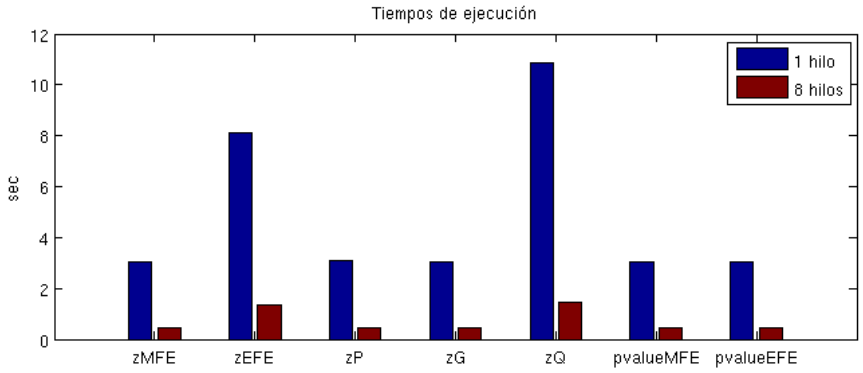


Figura 4.3: tiempos de ejecución antes y después de la paralelización.

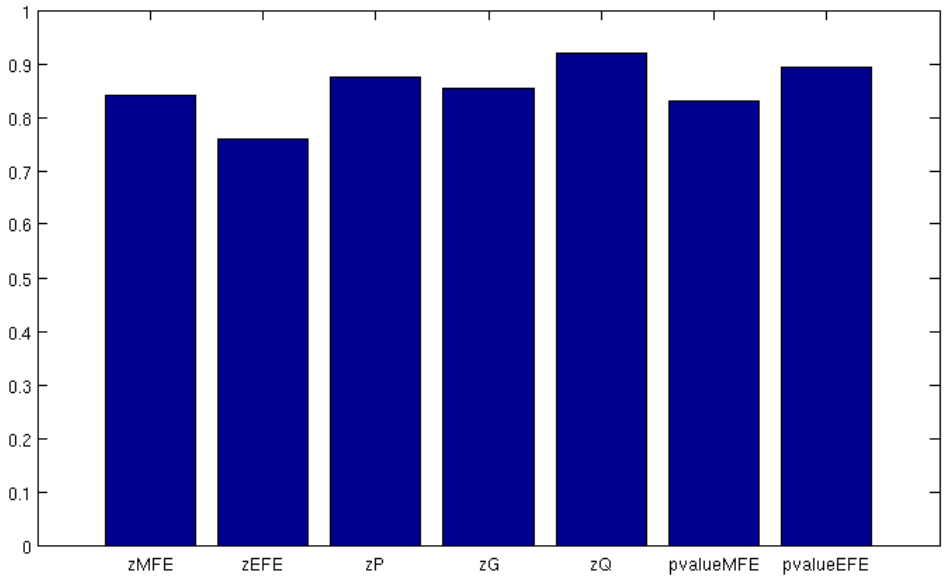


Figura 4.4: eficiencia de la paralelización

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

La biblioteca desarrollada permite crear sistemas de clasificación de miRNA de forma más simple, ya que resuelve una gran parte del proceso previo de extracción de características. La interfaz gráfica de usuario da la posibilidad a investigadores con poca experiencia en programación, pasando por un proceso de aprendizaje mínimo, de aprovechar las funciones de la biblioteca. La posibilidad de leer secuencias en formato *fasta* y guardar resultados en formato *csv* hace que el software sea compatible con la mayor parte del software utilizado en el campo.

Cabe señalar también que la revisión sistemática realizada sobre las características utilizadas en el estado del arte, además de cumplir con uno de los objetivos planteados al inicio del trabajo, es un producto secundario del proyecto que tiene valor propio y no es sólo un paso intermedio de éste.

Los resultados de la biblioteca fueron comparados con los de otros autores, por lo se puede confiar en que los procesos están libres de errores y siguen correctamente las especificaciones originales. Además, las pruebas de desempeño arrojaron tiempos de ejecución del orden de los milisegundos en la mayoría de las funciones, lo que habilita el análisis de grandes volúmenes de datos sin tiempos excesivos de espera.

Asimismo, los resultados de la paralelización fueron muy positivos, más aún si se considera que el microprocesador contaba solo con 4 núcleos reales, y que los restantes hilos de ejecución son virtuales.

Se espera que el aporte realizado con este proyecto facilite el estudio de la utilidad y la relevancia de las características utilizadas en el estado del arte, además de permitir la experimentación con distintos tipos de clasificadores y métodos de entrenamiento.

5.2. Trabajos futuros

Como se detalló al comienzo del documento, el alcance de este proyecto se limitaba al desarrollo de una biblioteca de extracción de características. El próximo paso lógico es la implementación de un clasificador que, aprovechando los descriptores que brinda la biblioteca, construya una base de conocimiento para luego clasificar secuencias de miRNA.

Cabe señalar que no se implementaron procesos de extracción de algunas características. El motivo principal es que los algoritmos son demasiado complejos y no se encontró un software libre que los implemente. A futuro se pretende implementar estos métodos internamente en la biblioteca. La lista de las características que quedaron fuera de la biblioteca y las fuentes bibliográficas correspondientes se presentan en el Anexo A.

Apéndice A

Características que no se implementaron

- **Distancia de pares de bases ajustada (dD)**: [24] y [28]
- **Segundo valor propio o valor propio de Fiedler (dF)**: [24], [28] y [36]
- **Distancia de pares de bases ajustada normalizada (zD)**: [24], [28] y [36]
- **Segundo valor propio o valor propio de Fiedler normalizada (zF)**: [24], [28] y [36]
- **Entropía de la estructura (dS)**: [28], [16] y [36]
- **Entalpía de la estructura (dH)**: [28], [16] y [36]
- **Energía de fusión de la estructura (Tm)**: [28], [16] y [36]
- **Entropía de la estructura ajustada (dS/l)**: [28], [16] y [36]
- **Entalpía de la estructura ajustada (dH/l)**: [28], [16] y [36]]
- **Energía de fusión de la estructura ajustada (Tm/l)**: [28], [16] y [36]
- **Estabilidad**: [11]
- **Estabilidad alternativa**: [11]
- **Características orientadas a precursores de múltiple bucle**: [16]

Bibliografía

[1] Bartel, DP: *MicroRNAs: genomics, biogenesis, mechanism, and function.* Cell, 116 (2):281–297, 2004.

[2] Bartlett, JMS y D Stirling: *A Short History of the Polymerase Chain Reaction.* PCR Protocols, 226:3–6, 2003.

[3] Bentwich, I, A Avniel, Y Karov, R Aharonov, S Gilad, O Barad, A Barzilai, P Einat, U Einav, E Meiri, E Sharon, Y Spector y Z Bentwich: *Identification of hundreds of conserved and nonconserved human microRNAs.* Nat Genet, 37(7):766–770, 2005.

[4] Bonnet, E, J Wuyts, P Rouzé y Y Van de Peer: *Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences.* Bioinformatics, 20 (17):2911–2917, 2004.

[5] Chenghai, X, L Fei, H Tao, L Guo-Ping, L Yanda y Z Xuegong: *Classification of real and pseudo microRNA precursors using local structure sequence features and support vector machine.* BMC Bioinformatics, 6:310, 2005.

[6] Curtis, H, NS Barnes, A Schnek y A Massarini: *Biología.* Editorial Médica Panamericana Buenos Aires, 2008.

[7] Dayhoff, MO, RM Schwartz y BC Orcutt: *In Atlas of Protein Sequence and Structure*, volumen 5 (3). 1978.

[8] Dua, S y P Chowriappa: *Data Mining For Bioinformatics.* CRC Press, 2012.

[9] Esquela-Kerscher, A y FJ Slack: *Oncomirs - microRNAs with a role in cancer.* Nature Reviews Cancer, 6(1):259–269, 2006.

[10] Goro, T, K Takashi, A Kiyoshi y K Taishin: *miRRim: A novel system to find conserved miRNAs with high sensitivity and specificity.* RNA, 13 (12):2081–2090, 2007.

[11] Hackenberg, M, M Sturm y D Langenberger: *miRanalyzer: a microRNA detection and analysis tool for next-generation sequencing experiments.* Nucleic Acids Research, 37:68–76, 2009.

[12] Henikoff, S y JG Henikoff: *Amino acid substitution matrices from protein blocks.* Proc Natl Acad Sci USA, 89 (22):10915–9, 1992.

[13] Hertel, J y PF Stadler: *Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data.* Bioinformatics, 22 (14):e197–e202, 2006.

[14] Hofacker, IL: *Vienna RNA secondary structure server.* Nucleic Acids Res., 31:3429–3431, 2003.

[15] Huang, TH, B Fan, M Rothschild, ZL Hu, K Li y SH Zhao: *MiRFinder: an improved approach and software implementation for genome-wide fast microRNA precursor scans.* BMC Bioinformatics, 8(1):341, 2007.

[16] Jiandong, D, Z Shuigeng y G Jihong: *MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features.* BMC Bioinformatics, 11 (11):11, 2010.

[17] Jopling, C, M Yi, A Lancaster, S Lemon y P Sarnow: *Modulation of hepatitis C virus RNA abundance by a liver-specific MicroRNA.* Science, 309(5740):1577–1581, 2005.

[18] Lecellier, CH, P Dunoyer, K Arar, J Lehmann-Che, S Eyquem, C Himber, A Saib y O Voinnet: *A cellular MicroRNA mediates antiviral defense in human cells.* Science, 308:557–560, 2005.

[19] Lee, RC, RL Feinbaum y V Ambros: *The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14.* Cell, 75(5):843–854, 1993.

[20] Li, L, J Xu, D Yang, X Tan y H Wang: *Computational approaches for microRNA studies: a review.* Mamm Genome, 21(1):1–12, 2010.

[21] Lim, LP, NC Lau, EG Weinstein, A Abdelhakim, S Yekta, MW Rhoades, CB Burge y DP Bartel: *The microRNAs of Caenorhabditis elegans.* Genes & development, 17(8):991–1008, 2003.

[22] McCaskill, JS: *The equilibrium partition function and base pair binding probabilities for RNA secondary structure.* Biopolymers, 29:1105–1119, 1990.

[23] Michael, H, H Florence, L Yuan, D Gregory y S Belinda: *NIH WORKING DEFINITION OF BIOINFORMATICS AND COMPUTATIONAL BIOLOGY*, 2008.

[24] Ng, KLS y SK Mishra: *De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures.* Bioinformatics, 23(11):1321–30, 2007.

[25] Peng, J, W Haonan, W Wenkai, M Wei, S Xiao y L Zuhong: *MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features.* Nucleic Acids Research, 35:339–44, 2007.

[26] Pevsner, J: *Bioinformatics and functional genomics.* Wiley Blackwell, 2009.

[27] Rosenzvit, M, M Cucher, L Kamenetzky, N Macchiaroli, L Prada y F Camicia: *MicroRNAs in Endoparasites.* Nova Science Publishers, páginas 65–92, 2013.

[28] Rukshan, B y P Vasile: *microPred: effective classification of pre-miRNAs for human miRNA gene prediction.* Bioinformatics, 25(8):989–995, 2009.

[29] Sewer, A, N Paul, P Landgraf, A Aravin, S Pfeffer, MJ Brownstein, T Tuschl, E van Nimwegen y M Zavolan: *Identification of clustered microRNAs using an ab initio prediction method.* BMC Bioinformatics, 6:267, 2005.

[30] Siepel, A y D Haussler: *Phylogenetic hidden Markov models.* En *In statistical methods in molecular evolution*, páginas 325–351. Springer, 2005.

[31] Snorre, AH, S Ola y S Pal: *Reliable prediction of Drosha processing sites improves microRNA gene prediction.* Bioinformatics, 23(2):142–149, 2007.

[32] Thompson, JD, DG Higgins y TJ Gibson: *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.* Nucleic Acids Research, 22 (22):4673–4680, 1994.

[33] Voysest, O: *Mejoramiento genético del frijol (Phaseolus vulgaris L): legado de variedades de América Latina 1930-1999.* Número 321. Free download form CIAT, 2000.

[34] Workman, C y A Krogh: *No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution.* Nucleic Acids Research, 27 (24):4816–22, 1999.

[35] Wuchty, S, W Fontana, IL Hofacker y P Schuster: *Complete suboptimal folding of RNA and the stability of secondary structures.* Biopolymers, 49(2):145–65, 1999.

[36] Xuan, P, MZ Guo, J Wang, CY Wang, XY Liu y Y Liu: *Genetic algorithm-based efficient feature selection for classification of pre-miRNAs.* Genet. Mol. Res., 10 (2):588–603, 2011.

[37] Xue, C, F Li, T He, GP Liu, Y Li y X Zhang: *Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine.* BMC Bioinformatics, 6(1):310, 2005.

[38] Yousef, M, M Nebozhyn, H Shatkay, S Kanterakis, LC Showe y MK Showe: *Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier.* Bioinformatics, 22 (11):1325–1334, 2006.

[39] Zhang, BH, XP Pan, SB Cox, GP Cobb y TA Anderson: *Evidence that miRNAs are different from other RNAs.* Cell. Mol. Life Sci., 63(2):46–254, 2006.

[40] Ziheng, Y: *A Space-Time Process Model for the Evolution of DNA Sequences.* Genetics, 139:993–1005, 1994.

[41] Zuker, M y P Stiegler: *Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information.* Nucl Acid, 9:133–148, 1981.