UNIVERSIDAD NACIONAL DEL LITORAL

DOCTORADO EN INGENIERÍA

Nuevos enfoques basados en medidas de complejidad para la detección de secuencias cortas en bioinformática

Jonathan Raad

FICH
FACULTAD DE INGENIERÍA
Y CIENCIAS HÍDRICAS

sinc(i)
INSTITUTO DE INVESTIGACIÓN EN SEÑALES
SISTEMAS E INTELIGENCIA COMPUTACIONAL

INTEC
INSTITUTO DE DESARROLLO TECNOLÓGICO
PARA LA INDUSTRIA QUÍMICA

CIMEC
CENTRO DE INVESTIGACIÓN DE
MÉTODOS COMPUTACIONALES

Tesis de Doctorado **2021**

UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional

NUEVOS ENFOQUES BASADOS EN MEDIDAS DE COMPLEJIDAD PARA LA DETECCIÓN DE SECUENCIAS CORTAS EN BIOINFORMÁTICA

Jonathan Raad

Tesis remitida al Comité Académico del Doctorado como
parte de los requisitos para la obtención del grado de
DOCTOR EN INGENIERÍA
Mención en Inteligencia Computacional, Señales y Sistemas
de la
UNIVERSIDAD NACIONAL DEL LITORAL

2021

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria, Paraje "El Pozo", S3000,
Santa Fe, Argentina.

UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional

NUEVOS ENFOQUES BASADOS EN MEDIDAS DE COMPLEJIDAD PARA LA DETECCIÓN DE SECUENCIAS CORTAS EN BIOINFORMÁTICA

Jonathan Raad

Lugar de Trabajo:
sinc(i)

Instituto de Señales, Sistemas e Inteligencia Computacional
Facultad de Ingeniería y Ciencias Hídricas
Universidad Nacional del Litoral

Director:
Dr. Diego H. Milone — sinc(i), CONICET-UNL

Co-director:
Dra. Georgina Stegmayer — sinc(i), CONICET-UNL

Jurado Evaluador:
Dr. Leandro Lucero — IAL, CONICET-UNL
Dra. Elizabeth Tapia — CIFASIS, CONICET-UNR
Dra. Jéssica Carballido — ICIC, CONICET-UNS

UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Santa Fe, 17 de Diciembre de 2021.

Como miembros del Jurado Evaluador de la Tesis de Doctorado en Ingeniería titulada ***"Nuevos enfoques basados en medidas de complejidad para la detección de secuencias cortas en bioinformática"***, desarrollada por el Bioing. Jonathan RAAD, en el marco de la Mención "Inteligencia computacional, señales y sistemas", certificamos que hemos evaluado la Tesis y recomendamos que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

La aprobación final de esta disertación estará condicionada a la presentación de dos copias encuadernadas de la versión final de la Tesis ante el Comité Académico del Doctorado en Ingeniería.

Dra. Elizabeth Tapia	Dr. Leandro Lucero	Dra. Jessica Carballido

Santa Fe, 17 de Diciembre de 2021.

Certifico haber leído la Tesis, preparada bajo mi dirección en el marco de la Mención "Inteligencia computacional, señales y sistemas" y recomiendo que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

.....	.....
Dra. Georgina Stegmayer	Dr. Diego Milone
Codirectora de Tesis	Director de Tesis

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS * SANTA FE *

Dr. JOSÉ LUIS MACOR
SECRETARIO DE POSGRADO
Facultad de Ingeniería y Cs. Hídricas

Universidad Nacional del Litoral
Facultad de Ingeniería y
Ciencias Hídricas

Secretaría de Posgrado

Ciudad Universitaria
C.C. 217
Ruta Nacional Nº 168 – Km. 472,4
(3000) Santa Fe
Tel: (54) (0342) 4575 229
Fax: (54) (0342) 4575 224
E-mail: posgrado@fich.unl.edu.ar

DECLARACIÓN LEGAL DEL AUTOR

Esta Tesis ha sido remitida como parte de los requisitos para la obtención del grado académico de Doctor en Ingeniería ante la Universidad Nacional del Litoral y ha sido depositada en la Biblioteca de la Facultad de Ingeniería y Ciencias Hídricas para que esté a disposición de sus lectores bajo las condiciones estipuladas por el reglamento de la mencionada Biblioteca.

Se permiten citaciones breves de esta Tesis sin la necesidad de un permiso especial, en la suposición de que la fuente sea correctamente citada. El portador legal del derecho de propiedad intelectual de la obra concederá por escrito solicitudes de permiso para la citación extendida o para la reproducción parcial o total de este manuscrito.

TESIS POR COMPILACIÓN

La presente tesis se encuentra organizada bajo el formato de Tesis por Compilación, aprobado en la resolución No 255/17 (Expte. No 888317-17) por el Comité Académico de la Carrera Doctorado en Ingeniería, Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral (UNL). De dicha resolución:

"En el caso de optar por la Tesis por Compilación, ésta consistirá en una descripción técnica de al menos 30 páginas, redactada en español e incluyendo todas las investigaciones abordadas en la tesis. Se deberán incluir las secciones habituales indicadas a continuación en la Sección Contenidos de la Tesis. Los artículos científicos publicados por el autor, en el idioma original de las publicaciones, deberán incluirse en un Anexo con el formato unificado al estilo general de la Tesis indicado en la Sección Formato. El Anexo deberá estar encabezado por una sección donde el tesista detalle para cada una de las publicaciones cuál ha sido su contribución. Esta sección deberá estar avalada por su director de Tesis. El documento central de la Tesis debe incluir referencias explícitas a todas las publicaciones anexadas y presentar una conclusión que muestre la coherencia de dichos trabajos con el hilo conceptual y metodológico de la tesis. Los artículos presentados en los anexos podrán ser artículos publicados, aceptados para publicación (en prensa) o en revisión."

AGRADECIMIENTOS

A mis directores, Dr. Diego H. Milone y Dra. Georgina Stegmayer, por creer en mí y darme la confianza y la oportunidad de formar parte de su grupo de investigación. Su compromiso por guiarme en los primeros pasos de mi carrera científica, su paciencia, dedicación y pasión por su trabajo me han enseñado que el éxito es la consecuencia de amar lo que uno elige hacer.
A mis compañeros y colegas del Instituto sinc(i). Más allá del vasto conocimiento técnico compartido, las excelentes relaciones humanas me han permitido transitar un camino ameno durante mis años como doctorando.
A mis padres y hermana por el apoyo incondicional que me han dado durante toda mi vida, motivándome siempre a dar lo mejor de mi.
A Dios por darme el don de vida.
Finalmente, quiero agradecer a las siguientes instituciones:

- sinc(i): Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional.
- Facultad de Ciencias Hídricas de la Universidad Nacional del Litoral.
- CONICET: Consejo Nacional de Investigaciones Científicas y Técnicas.

Jonathan Raad
Santa Fe, Diciembre de 2021.

Índice general

Índice de tablas

Índice de figuras

Resumen

El aprendizaje maquinal ha tenido un gran desarrollo en los últimos años y ha permitido resolver una gran cantidad de problemas en las más diversas disciplinas. Sin embargo, aún quedan grandes desafíos, como lo es el aprendizaje en datos con alto grado de desbalance de clases o con muy pocos datos etiquetados. Un caso particular de aplicación donde se presentan desafíos como estos es en la predicción computacional de secuencias de microARN (miARN). El miARN, también llamado miARN maduro, es una pequeña molécula de ácido ribonucleico (ARN) no codificante la cual puede regular la expresión de los genes en la célula.

En los últimos años, se ha desarrollado una gran cantidad de métodos que intentan detectar nuevos miARN utilizando información principalmente de su estructura. El primer paso en estos métodos generalmente consiste en extraer del genoma subcadenas de nucleótidos que cumplan con ciertos requerimientos estructurales. En segundo lugar se extraen características numéricas de estas subcadenas. Finalmente se utiliza el aprendizaje maquinal para predecir cuáles de estas secuencias de nucleótidos pueden codificar un miARN. El principal inconveniente de estos métodos es que utilizan características basadas principalmente en la estructura del precursor (pre-miARN) sin incluir la información del miARN maduro, que se encuentra codificada en forma secuencial. De esta manera, se pierde información muy valiosa que podría utilizarse para mejorar la predicción de nuevos pre-miARN y disminuir a su vez el número de falsos positivos.

Más recientemente, en varios dominios de aplicación se propusieron los enfoques basados en aprendizaje profundo como un método para la extracción automática de características. Sin embargo, éstos tienen aún importantes limitaciones prácticas cuando deben aplicarse a tareas de predicción real. En primer lugar requieren de un extenso preprocesamiento de los datos. En segundo lugar, para el caso de predicción de pre-miRNA requieren del uso de modelos externos no neuronales (como RNAfold) para poder realizar el cálculo de la estructura secundaria. Finalmente, existe el desafío de lidiar con un número escaso de ejemplos de pre-miARN positivos y con el alto desbalance de clases (que puede llegar hasta 1:160.000) en los genomas completos.

Para permitir la predicción de nuevos miARNs en genomas completos disminuyendo la cantidad de falsos positivos, en esta tesis se realizaron dos grandes aportes. En primer lugar, se desarrollaron tres nuevas características basadas en medidas de complejidad del miARN maduro, las cuales permiten reducir significativamente el número de falsos positivos en los clasificadores, manteniendo un alto desempeño aún para desbalances extremos. En segundo lugar, se desarrolló el primer algoritmo de aprendizaje profundo de extremo a extremo para la predicción de pre-miARNs en genomas completos, el cual no requiere de ningún pre-procesamiento de los datos ni de modelos externos para el cálculo de la estructura secundaria. Este algoritmo fue realizado con Transformers, el cual es un nuevo modelo de aprendizaje profundo basado en mecanismos de atención. Así, es posible extraer de manera automática la información necesaria para estimar la estructura secundaria y la mínima energía libre de cada secuencia, mejorando la predicción de nuevos pre-miARNs y disminuyendo la cantidad de falsos positivos para casos de predicción en genomas completos.

Sección 1

Introducción

1.1. Medidas de complejidad

En las últimas décadas el desarrollo de medidas de complejidad (MC) han permitido el análisis de sistemas complejos que no pueden ser fácilmente modelados. Estas medidas pueden evaluar la complejidad de un sistema a partir del análisis de las secuencias temporales producidas por los mismos Grassberger (1991). A finales de los años 90, diferentes autores introdujeron el uso de las MC para el análisis y detección de patrones en secuencias biológicas Nan and Adjeroh (2004); D. Gusev *et al.* (1999). Desde entonces, se abordaron exitosamente distintos problemas bioinformáticos con el uso de MC. Un problema de especial interés en bioinformática es el descubrimiento de microARN (miARN). El miARN, también llamado miARN maduro, es una pequeña molécula de ARN no codificante de aproximadamente 21 nucleótidos de largo, que regula la expresión de los genes en la célula Bartel (2018). Estas moléculas se encuentran codificadas en los precursores de miARN (pre-miARN), que son secuencias de aproximadamente unos 100 nucleótidos de largo. Estos precursores, luego de ser transcritos como ARN, se pliegan sobre sí mismos formando estructuras secundarias del tipo tallo-bucle, también llamadas horquillas (ver Figura 1.1). Debido a su rol clave en la promoción e inhibición de genes y su importancia en diversas enfermedades, el descubrimiento de nuevos pre-miARNs es actualmente de gran interés Chen *et al.* (2018).

Para encontrar nuevos pre-miARNs se entrena un clasificador de modo supervisado a partir de características secuenciales y estructurales extraídas de las secuencias de un genoma (Li *et al.*, 2010; de ON Lopes *et al.*, 2014; Shukla *et al.*, 2017). En la literatura han sido propuestos varios conjuntos de características diferentes, los cuales principalmente describen información de la estructura de un pre-miARN. Éstas son mayormente inspiradas en la acción de la enzima Drosha (de ON Lopes *et al.*, 2014), la cual puede reconocer horquillas que tengan propiedades estructurales específicas de un pre-miARN. Sin embargo, aunque esta enzima toma un rol principal en la selección de aquellas horquillas precursoras de miARNs, la especificidad de los subsiguientes procesos puede imponer restricciones adicionales a aquellas horquillas que eventualmente se convertirán en miARN maduros (Bartel, 2018). En diferentes estudios se ha encontrado que la selección de los ARN mensajeros que son objetivo de miARN se define por la secuencia de su correspondiente miARN maduro (Friedman *et al.*, 2009; Lewis *et al.*, 2005; Brennecke *et al.*, 2005; Bartel, 2009). Específicamente, el miARN maduro contiene dos áreas de unión con la secuencia objetivo llamadas semilla y sitio complementario (Friedman *et al.*, 2009). Dada la importancia que tiene la semilla en la función de la secuencia, los miARN maduros que comparten la misma semilla pueden ser clasificados en grupos llamados familias de miARN (Lewis *et al.*, 2003).

Dado entonces que la información importante está codificada en la región del maduro de un pre-miARN, la estructura secundaria del precursor en sí misma podría no ser suficiente para diferenciar un pre-miARN verdadero de otras horquillas. Nuestra hipótesis es que la principal dificultad para separar ambas clases se debe a la omisión de información relevante sobre la secuencia del miARN maduro en el proceso de extracción de características de los pre-miARN. Este hecho es especialmente notable en la predicción de nuevos precursores, donde las características se extraen principalmente de

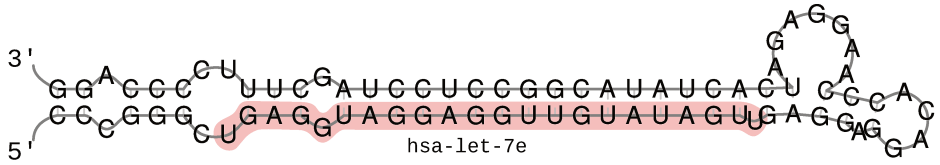

Figura 1.1: Estructura secundaria de tipo horquilla de un pre-miARN. Se indica resaltado su miARN maduro.

su estructura. Un ejemplo típico de este tipo de características estándar (CE) es la representación de tripletes (Xue *et al.*, 2005), que considera la composición estructural de tres nucleótidos adyacentes y la base media para construir un vector con la frecuencia de aparición de 32 posibles elementos. Otros ejemplos son el número de bucles internos y su longitud (Yousef *et al.*, 2006), la puntuación z de la energía libre mínima (Hertel and Stadler, 2006), la proporción de dinucleótidos (Batuwita and Palade, 2009), la proporción de pares de bases en el tallo, el contenido de guanina y citosina (G + C) en el bucle terminal (de ON Lopes *et al.*, 2014), la entropía de Shannon (zQ), la propensión de pares de bases (zP) (Ng and Mishra, 2007) y la distancia entre pares de bases (zD) (Ding *et al.*, 2010). Aunque se han propuesto muchas características, éstas se basan principalmente en la estructura secundaria del pre-miARN o las frecuencias relativas de dinucleótidos, trinucleótidos y motivos en estas secuencias (de ON Lopes *et al.*, 2014; Yones *et al.*, 2015). Estas características se han desempeñado bastante bien en los clasificadores actuales (Stegmayer *et al.*, 2018), pero sin embargo se puede afirmar que las CE no permiten representar ni preservar la información sobre el orden en que estas tríadas y motivos están presentes en la secuencia, perdiendo información valiosa sobre la codificación del miARN maduro dentro de la secuencia del precursor. Para esto, en una primera parte de la tesis proponemos tres nuevas características que tienen especialmente en cuenta el orden en el que se presentan los nucleótidos en el miARN maduro. Mostraremos cómo estas características novedosas mejoran efectivamente la predicción de nuevos pre-miARN, independientemente del clasificador.

Una de las características propuestas se basa en la distancia de Levenshtein. La razón fundamental de esta propuesta es que las secuencias candidatas, para ser nuevos miARN, deben ser muy similares en la región que codifica el maduro, y la distancia de Levenshtein puede medirlo en términos de cantidad de ediciones de nucleótidos entre un par de secuencias. Esta distancia se ha utilizado en otras áreas de la bioinformática como en la alineación de secuencias, y también para estimar la proximidad entre secuencias (Zytnicki *et al.*, 2008; Lassmann and Sonnhammer, 2005; Billoud *et al.*, 2013). Por tanto, debido a la conservación y evolución de los miARN (Wheeler *et al.*, 2009), mostraremos cómo las cadenas que codifican el miARN maduro de posibles pre-miARN están más próximas en este espacio que las que no codifican un miARN. De esta manera es posible calcular, para cada secuencia, una distancia de Levenshtein a los pre-miARN conocidos (clase positiva) con el fin de evaluar qué tan cerca está cada secuencia candidata de los pre-miARN conocidos. A diferencia de lo propuesto por (Mathelier and Carbone, 2010), donde la distancia de Levenshtein se utiliza como un cálculo directo de los errores de edición con un umbral para eliminar secuencias como primer paso del procesamiento, en nuestro trabajo construimos un estadístico que puede estimar la pertenencia de una secuencia al conjunto de ejemplos de la clase positiva. De esta forma, la distancia de Levenshtein como característica es más general y aplicable a cualquier especie, y puede ser utilizada por cualquier clasificador.

La segunda y tercera característica propuesta se inspiraron desde el punto de vista de la teoría de la información, considerando la aleatoriedad de una secuencia que codifica un miARN maduro en la horquilla. Además, se sabe que ciertas regiones maduras tienen motivos específicos que definen su funcionalidad y la pertenencia a una familia de miARN específica (Bartel, 2018, 2009). Para poder cuantificar esto proponemos las características de entropía de permutación (Bandt and Pompe, 2002) y complejidad de Lempel-Ziv (Ziv and Lempel, 1978).

1.2. Extracción de características con aprendizaje profundo

En varios dominios bioinformáticos el gran desafío actual es el desarrollo de métodos de aprendizaje maquinal que no requieran ningún pre-procesamiento de la entrada, es decir, lo que se suele denominar modelo de extremo a extremo (Trieu *et al.*, 2020; Tsubaki *et al.*, 2018; Chaabane *et al.*, 2019). En el escenario de predicción de pre-miARNs en genomas completos, dicho método debería poder entrenarse solo con secuencias de ARN crudas (una directa codificación numérica de los aminoácidos), para luego ser capaz de recibir el genoma crudo de cualquier especie sin extracción de características ni cálculo de estructura secundaria. Sin embargo, dado que en tal escenario no es posible descartar previamente aquellas secuencias que no se pliegan como horquillas (las que denominaremos planas), es necesario incorporarlas también al entrenamiento. Precisamente, para evitar cualquier paso de la ingeniería de características, la aparición del aprendizaje profundo (AP) ha provisto mejoras significativas en el campo de la representación automática para la visión por computadora, el reconocimiento de voz y muchos otros dominios de aplicaciones (LeCun *et al.*, 2015). Los modelos profundos pueden extraer automáticamente características relevantes por sí mismos, directamente a partir de datos sin procesar, por lo que se consideran hoy en día el mejor paradigma de aprendizaje maquinal para la mayoría de las tareas de clasificación (Jurtz *et al.*, 2017; Bengio *et al.*, 2013). En bioinformática el AP ya se ha utilizado para la extracción, identificación y clasificación de características de ARN pequeños (Zheng *et al.*, 2019; Amin *et al.*, 2019; Zeng *et al.*, 2016). Además, el AP puede detectar motivos en un conjunto de secuencias homólogas, que luego son la clave para distinguir entre diferentes tipos de familias de proteínas o predecir su estructura (Seo *et al.*, 2018; Senior *et al.*, 2020). En (Eraslan *et al.*, 2019) los autores analizan las lagunas y los desafíos del AP en genómica, y mencionan la necesidad de más herramientas basadas en AP, capaces de manejar el escenario real de todo el genoma con modelos completos de extremo a extremo, sin requerir ningún tipo de pre-procesamiento manual o ingeniería de características.

Recientemente se ha propuesto un modelo basado en redes neuronales convolucionales (CNN), denominado deepMir, para la clasificación de familias de miARN (Tang and Sun, 2019). A diferencia de la mayoría de las herramientas de clasificación binaria, el enfoque aquí es clasificar las secuencias de entrada en diferentes familias de miARN para una anotación de funciones más detallada. Este modelo recibe como entrada sólo secuencias de ARN utilizando un esquema de codificación de ceros y unos (one-hot-encoding). Así, convierte una secuencia de ARN en una matriz a la entrada de la red, codificando de esta manera los 4 tipos de nucleótidos en la secuencia. El modelo de CNN contiene dos capas convolucionales, seguidas de capas de pooling y tres capas completamente conectadas con dropout. En (Bugnon *et al.*, 2020a) se demostró que el rendimiento de deepMir, utilizado en predicción de pre-miRNA, estaba por debajo de los modelos profundos que utilizan también la estructura secundaria predicha como entrada, como deepMiRGene (Park *et al.*, 2017). Sin embargo, deepMir es un paso importante hacia modelos totalmente entrenables a partir de secuencias genómicas sin procesar y un punto de partida para lograr modelos de extremo a extremo, con el potencial de superar a otros enfoques gracias a la capacidad de extraer las características automáticamente.

Como alternativa para mejorar los modelos AP en la extracción automática de características, los modelos denominados Transformers han aparecido muy recientemente, provenientes del dominio del procesamiento del lenguaje natural (Devlin *et al.*, 2018a; Vaswani *et al.*, 2017). Los Transformers son redes profundas con mecanismos de auto-atención en cada capa, lo que permite obtener varias mejoras con respecto a los modelos recurrentes y convolucionales (Dosovitskiy *et al.*, 2020). Por un lado, el flujo de información se paraleliza, en lugar de realizarse de forma secuencial como en las redes recurrentes. Por otro lado, a diferencia de las redes convolucionales que trabajan con una visión local y requieren de muchas capas para obtener una visión global, los mecanismos de atención permiten el análisis de secuencias más largas sin perder información de contexto y manteniendo una visión global de la entrada en cada capa gracias a sus conexiones punto a punto (Vaswani *et al.*, 2017). Estas características de los Transformers les pueden permitir aprender relaciones entre todos los nucleótidos dentro de una secuencia en forma de horquilla, pudiendo así modelar su estructura secundaria. De esta manera, es posible desarrollar un modelo de aprendizaje profundo capaz de extraer información sólo a partir de la secuencia de ARN, sin ningún preprocesamiento de datos.

Por lo tanto, en la segunda parte de esta tesis buscamos analizar directamente el ADN con he-

rramientas de procesamiento del lenguaje natural (PLN) y aprendizaje profundo. Este enfoque está orientado a poder clasificar un candidato a miARN solo a partir de su secuencia y su contexto en el genoma sin utilizar ningún pre-procesamiento ni extracción de características. Así, en esta etapa se desarrollaron arquitecturas profundas basadas en Transformers y mecanismos de atención, dado que éstas han mostrado ser capaces de analizar grandes contextos en problemas de PLN (Vaswani *et al.*, 2017). De esta manera, se busca eliminar el sesgo producido por las estructuras ya conocidas de tipo horquilla y permitir la detección de nuevos candidatos que no pertenezcan a las familias ya conocidas de miARNs.

1.3. Predicción automática de miARN

Las técnicas experimentales para la predicción de pre-miARNs involucran diferentes dificultades tecnológicas y prácticas, por lo cual los métodos computacionales han ido ocupando un rol cada vez más preponderante en los últimos 10 años. Sin embargo, y a pesar de los esfuerzos de la comunidad científica, los métodos computacionales propuestos aún tienen muchos desafíos por resolver Stegmayer *et al.* (2018). Esto es debido a la gran cantidad de falsos positivos generados por los clasificadores del estado del arte, lo que hace impráctica su aplicación. En general hay tres grandes inconvenientes con los enfoques aplicados hasta el momento. En primer lugar, existe un fuerte desbalance de datos en el conjunto de entrenamiento, lo que produce un sesgo hacia la clase mayoritaria (clase negativa), generando una disminución de la precisión, con muchos falsos positivos de la clase minoritaria (clase positiva, los pre-miARNs conocidos)(Bugnon *et al.*, 2020b). En segundo lugar, la mayoría de las características extraídas de los precursores son calculadas en función de la estructura de unos pocos ejemplos positivos, generando así un sesgo en el clasificador hacia unas pocas familias de precursores con estructuras conocidas. En tercer lugar, al utilizar sólo características estructurales se pierde información valiosa almacenada en la molécula del miARN maduro, codificada de manera secuencial. Por lo tanto, es fundamental poder desarrollar nuevos métodos computacionales que puedan capturar y representar mejor esta información secuencial, con especial interés en las regiones clave de la secuencia.

Actualmente existen distintos métodos de identificación de miARNs, los cuales se pueden clasificar como: i) enfoques experimentales mediante secuenciamiento; ii) métodos comparativos basados en la conservación; y iii) métodos basados en aprendizaje maquinal (Kleftogiannis *et al.*, 2013). Los dos primeros métodos requieren de experimentos de laboratorio, en donde la identificación está principalmente basada en miARNs que se expresan de forma abundante y que son específicos de una especie y de condiciones experimentales particulares. En comparación, los métodos computacionales basados en aprendizaje maquinal permiten hacer predicciones entre distintas especies sin requerir de costosos experimentos de laboratorio Chen *et al.* (2018). En los métodos computacionales básicamente se comienza extrayendo del genoma cadenas que al plegarse tengan una estructura y longitud similar a un pre-miARN conocido. Luego se extraen características de su secuencia y estructura para que puedan codificarse numéricamente y se las envía a un algoritmo de aprendizaje maquinal supervisado que fue entrenado con secuencias conocidas positivas y negativas, el cual determinará si las secuencias a clasificar son probables nuevos pre-miARNs o no.

1.4. Objetivo general

El objetivo general de esta tesis es el desarrollo de nuevos enfoques para la extracción automática de características para secuencias de ARN, de modo tal de mejorar significativamente las tasas de predicción de los clasificadores de secuencias de pre-miARNs.

1.5. Objetivos específicos

A continuación se detallan los objetivos específicos de la presente investigación:

- Desarrollar nuevos algoritmos para extracción automática de características, que contemplen la distancia de edición de las partes del maduro de un miARN.

- Desarrollar nuevos algoritmos para extracción automática de características, que contemplen la entropía como medida de aleatoriedad del maduro de un miARN.
- Desarrollar nuevos algoritmos para extracción automática de características, que contemplen la creación de diccionarios automáticos como medida de complejidad del maduro de un miARN.
- Desarrollar nuevos algoritmos de predicción basados en aprendizaje profundo, que contemplen la información del contexto de los pre-miARN en el genoma.
- Evaluar los algoritmos desarrollados incluyendo la predictividad positiva, para considerar el impacto de los falsos positivos en alto desbalance.
- Aplicar los modelos y algoritmos que se desarrollen en el campo de la bioinformática, en particular para la predicción/clasificación de nuevos pre-miARN en el genoma humano.
- Validar las propuestas con datos reales a través del trabajo multidisciplinario.

Sección 2

Métodos propuestos

2.1. Medidas de complejidad del miARN maduro

En la Figura 2.1 se muestra una metodología general para la predicción de nuevos miARNs con métodos computacionales tradicionales. Ésta se puede dividir en tres etapas, las cuales se describen a continuación.

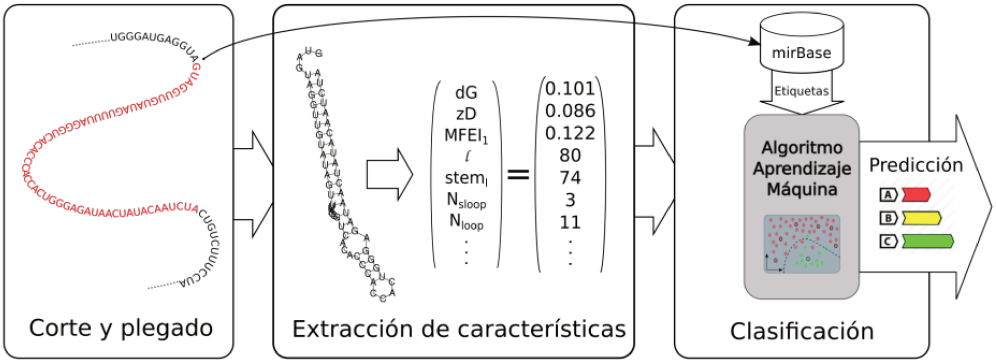

Figura 2.1: Etapas en la predicción de nuevos microARNs.

Para la detección de nuevos pre-miARNs es necesario un primer paso de "corte y plegado", el cual consiste en encontrar aquellas secuencias que son capaces de plegarse con una estructura secundaria en forma de horquilla. Este plegado permite medir distintas propiedades de la estructura de la molécula, como su longitud y nivel de emparejamiento de las bases. Estas propiedades son fundamentales como un primer filtrado para poder distinguir a las horquillas candidatas de las pseudo-horquillas en el genoma. Para realizar este procesamiento se requiere cortar el genoma en ventanas de longitud mayor a un pre-miARN y luego calcular su estructura secundaria. Finalmente, se seleccionan como posibles candidatos positivos aquellas horquillas que tienen una estructura similar a la de un pre-miARN.

El siguiente paso es la "extracción de características", que consiste en convertir las secuencias de nucleótidos (representadas como cadenas de caracteres) en vectores numéricos. De esta manera se pueden aplicar técnicas de aprendizaje maquinal para lograr separar las secuencias candidatas a pre-miARN de las que no lo son. Existen muchas características que pueden usarse para representar secuencias de ARN (Yones *et al.*, 2015; de ON Lopes *et al.*, 2014), pero la mayoría de ellas se encuentran basadas en propiedades de la estructura secundaria, lo cual puede no ser suficiente para diferenciar un pre-miARN de una pseudo-horquilla.

El último paso, la "clasificación" consiste en un algoritmo de aprendizaje maquinal que debe decidir si cada horquilla extraída previamente es un posible pre-miARN o no. Para entrenar este clasificador se

Algoritmo 1: Distancia de Levenshtein

Entrada: $\boldsymbol{x}$, $\boldsymbol{y}$ cadena de caracteres del ARN
Salida : L Distancia de Levenshtein

si $|\boldsymbol{x}||\boldsymbol{y}| = 0$ **entonces**
  $L \leftarrow \max\{|\boldsymbol{x}|, |\boldsymbol{y}|\}$
en otro caso
  $d_{i,0} \leftarrow i \ \ \forall i$
  $d_{0,j} \leftarrow j \ \ \forall j$
  para $i \leftarrow 1$ **a** $|\boldsymbol{x}|$ **hacer**
    para $j \leftarrow 1$ **a** $|\boldsymbol{y}|$ **hacer**
      $c \leftarrow 1 - \delta(x_i, y_j)$
      $d_{i,j} \leftarrow \min\{d_{i-1,j} + 1,\ d_{i,j-1} + 1,\ d_{i-1,j-1} + c\}$
  $L \longleftarrow d_{|\boldsymbol{x}|,|\boldsymbol{y}|} - ||\boldsymbol{x}| - |\boldsymbol{y}||$
devolver L

utilizan secuencias positivas obtenidas de la base de datos MirBase[1] y un conjunto de casos negativos generalmente conformado por secuencias que tienen una estructura de tipo horquilla pero se encuentran en regiones del genoma que son codificantes de proteínas.

En esta primera parte de la tesis se hicieron aportes a la etapa de extracción de características, incorporando información de la zona de la horquilla que codifica el miARN maduro, lo cual puede complementar la información de la estructura, permitiendo así mejorar la precisión en la predicción de nuevos pre-miARNs. Debido a que la información almacenada en el miARN maduro es secuencial y no estructural, se propusieron nuevas medidas para capturar, no solo la proporción de los nucleótidos, sino también el orden en el que los mismos están dispuestos. Para tal fin, se recurrió a medidas utilizadas en teoría de la información y alineamiento de secuencias, tales como la distancia de Levenshtein, entropía de permutación y la complejidad de Lempel-Ziv.

2.1.1. Distancia de Levenshtein

La distancia de Levenshtein (DL), L, también conocida como distancia de edición entre cadenas, se define como el mínimo número de operaciones (inserción, eliminación y/o sustitución) requeridas para transformar una cadena en otra (Levenshtein, 1966). Esta distancia entre dos cadenas $\boldsymbol{x}$ e $\boldsymbol{y}$, de largos $|\boldsymbol{x}|$ e $|\boldsymbol{y}|$, puede ser calculada mediante el Algoritmo 1.

Para calcular L como una característica para cada horquilla y dado que L es una distancia entre dos elementos, es necesario tener un conjunto de referencia para realizar la comparación. Sea $\mathcal{A}$ el conjunto con los miARNs maduros (clase positiva de referencia) $\boldsymbol{a}_k$. Sea $\boldsymbol{a}_\ell$ un elemento de $\mathcal{A}$ para el cual se desea obtener L. Entonces, la mediana de la distancia de $\boldsymbol{a}_l$ a todos los otros elementos del conjunto puede ser una característica de $\boldsymbol{a}_l$, esto es

$$L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \text{med}_{\forall k\neq\ell}\{\boldsymbol{a}_k, \boldsymbol{a}_\ell\}, \tag{2.1}$$

donde $\mathcal{A} \setminus \boldsymbol{a}_\ell$ es el conjunto $\mathcal{A}$ sin el elemento a_ℓ.

Debido a que cada candidato puede tener su maduro codificado en dos regiones distintas de la horquilla (5p y 3p), es necesario extraer dos cadenas $\boldsymbol{a}_\ell^{5p}$ y $\boldsymbol{a}_\ell^{3p}$. Esto se puede lograr con las dos medidas de cada $\boldsymbol{a}_\ell$, considerando como valor final a

$$L(\boldsymbol{a}_\ell) = \max\{L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p}), L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})\}. \tag{2.2}$$

[1] http://www.mirbase.org/

2.1.2. Entropía de permutación

La entropía de Shannon es ampliamente utilizada para medir la aleatoriedad de una secuencia. El inconveniente de este enfoque cuando se analizan secuencias de miARNs es que la información del orden en el que se disponen los nucleótidos se pierde al calcularse las frecuencias relativas. Para resolver esto, en (Bandt and Pompe, 2002) propusieron una nueva codificación basada en los patrones de permutaciones de una secuencia, donde la entropía es estimada a partir de las frecuencias relativas de esos patrones. La medida fue llamada entropía de permutación (EP). En este caso, la distribución de probabilidad de $\boldsymbol{x}$ fue reemplazada por las frecuencias relativas p_π de todos los posibles patrones π que pueden ser encontrados en $\boldsymbol{x}$. Así, cuando trabajamos con EP es necesario definir previamente la longitud de los patrones a ser permutados, lo que se denomina orden, N. Una vez definido el orden se pueden obtener $N!$ patrones π de largo N. Por ejemplo, seleccionando $N = 3$, se obtienen 6 patrones distintos: (1,2,3) (1,3,2) (2,1,3) (3,2,1) (3,1,2) (2,3,1). Si se calculan las frecuencias de esos patrones en $\boldsymbol{x}$, entonces la correspondiente EP puede ser estimada como

$$EP_N(\mathbf{x}) = -\sum_{i=1}^{N!} p_{\boldsymbol{\pi}_i} \cdot \log_2(p_{\boldsymbol{\pi}_i}). \tag{2.3}$$

2.1.3. Complejidad de Lempel-Ziv

El algoritmo de Lempel-Ziv permite el cálculo de la complejidad de una secuencia finita basada en el análisis de su "proceso de producción"(Ziv and Lempel, 1978). Bajo la hipótesis de que la secuencia de un maduro debe estar contenida en un diccionario y codificada solo por "palabras" específicas, se espera que los candidatos que codifican miARNs tengan un diccionario más pequeño que los candidatos que no lo tienen.

Sea $\boldsymbol{a}$ una secuencia de ARN, en la que se pueden alternar 4 nucleótidos diferentes. Definimos $\boldsymbol{a}(i, j)$ como una subsecuencia de $\boldsymbol{a}$ que se compone de los elementos que se encuentran entre los índices i y j. Decimos que a es reproducible desde $\mathbf{a}(1, j)$, si $\boldsymbol{a}(j + 1, |\boldsymbol{a}|)$ es una sub-palabra de $\boldsymbol{a}$ que está contenida en $\boldsymbol{a}(1, j)$. Luego, decimos que $\boldsymbol{a}$ es producible a partir de $\boldsymbol{a}(1, j)$ si se obtiene agregando al final de la secuencia $\boldsymbol{a}$ un nuevo elemento que no puede obtenerse a partir de la reproducción de $\boldsymbol{a}(1, j)$. En otras palabras, se puede obtener una cadena $\boldsymbol{a}$ a partir de la extensión de cadenas más pequeñas mediante dos procesos: reproducción, cuando la extensión se realiza copiando una subcadena de la cadena más pequeña; o producción, cuando la extensión se realiza mediante una nueva subcadena que no está contenida en la cadena inicial.

Si concatenamos todos los procesos por los cuales se puede formar la cadena $\boldsymbol{a}$, se obtiene la historia de su construcción, $H(\boldsymbol{a})$. Así, si consideramos cada paso del proceso como reproducción o producción, $\boldsymbol{a}$ puede analizarse como un proceso de z pasos $H(\boldsymbol{a}) = H_1(\mathbf{a})H_2(\mathbf{a})...H_z(\mathbf{a})$. Entonces, sea $|H(\mathbf{a})|$ el número de pasos en $H(\mathbf{a})$, la complejidad de Lempel-Ziv de una secuencia $\boldsymbol{a}$ se define como $\text{lz}(\boldsymbol{a}) = \text{mín}\{|H(\boldsymbol{a})|\}$, con respecto a todas las historias posibles de $\boldsymbol{a}$. Luego, para obtener una medida que sea independiente de la longitud de $\boldsymbol{a}$, se puede hacer

$$LZ(\boldsymbol{a}) = \frac{\text{lz}(\boldsymbol{a}) \log_4 |\boldsymbol{a}|}{|\boldsymbol{a}|} \tag{2.4}$$

donde 4 en la base del logaritmo representa el número de nucleótidos posibles.

2.1.4. Predicción de pre-miARN con las características propuestas

Para probar las características propuestas en una tarea de predicción real se realizó un entrenamiento de tipo supervisado con distintos clasificadores, de forma de validar que las características propuestas son útiles más allá del clasificador utilizado. Con este fin se utilizaron Naive Bayes (NB), Random Forest (RF), k-nearest neighbor (KNN) y Deep Belief Network (DBN) (Hinton *et al.*, 2006). Estos clasificadores fueron elegidos debido a que ellos han mostrado los mejores desempeños en un reciente trabajo en predicción de pre-miARNs (Stegmayer *et al.*, 2018).

Finalmente, hay que notar que debido a que la clase positiva solo cuenta con unos pocos miles de ejemplos etiquetados contra varias decenas de miles de ejemplos de la clase negativa, el conjunto de

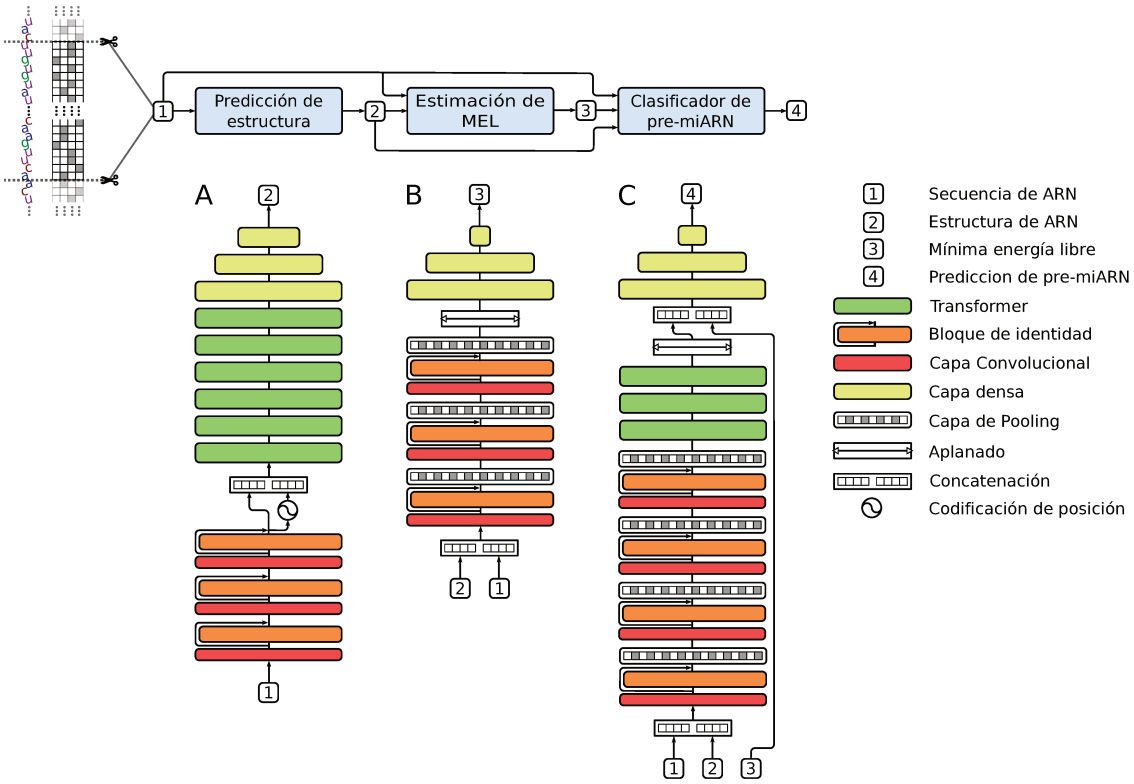

Figura 2.2: Representación esquemática del miRe2e completo (A) La secuencia de entrada de ARN 1 ingresa al modelo de predicción de estructura, que predice la estructura secundaria de plegamiento 2. (B) El modelo de estimación de MEL recibe 1 y 2 y calcula la energía libre mínima 3. (C) El modelo de clasificador de pre-miARN recibe 1, 2 y 3 para proporcionar la predicción de pre-miARN 4.

datos se encuentra altamente desbalanceado. Este desbalance produce un sesgo importante del clasificador hacia la clase mayoritaria, lo que genera un aumento de la cantidad de falsos positivos con una consecuente pérdida de la precisión. Por tal motivo, también se evaluó el desempeño de las características para distintos desbalances, con el objetivo de validar su comportamiento en una situación similar a la encontrada en un genoma real.

2.2. Predicción de pre-miARN con aprendizaje profundo

En esta parte de la tesis proponemos miRe2e, un modelo completo de aprendizaje profundo de extremo a extremo para la predicción de pre-miARN, basado en Transformers y mecanismos de atención (Vaswani *et al.*, 2017). Este modelo es capaz de recibir como entrada las secuencias de datos de todo el genoma sin ningún preprocesamiento. Después de un paso de entrenamiento con secuencias conocidas y sin etiquetar, puede identificar secuencias de pre-miARN dentro de un genoma completo. Este modelo aprende automáticamente las características estructurales intrínsecas de los precursores de un pre-miARN a partir de los datos sin procesar, es decir, sin ninguna ingeniería de características. La propuesta se probó con varias configuraciones experimentales con todo el genoma humano y se ha comparado con los algoritmos del estado del arte.

Dado que el miRe2e está diseñado para funcionar en todo el genoma sin ningún preprocesamiento, las secuencias de entrada se obtienen mediante un escaneo y corte del genoma con ventanas super-

puestas, las cuales tienen una longitud L y un paso s. Luego, cada secuencia se representa como un tensor de ceros y unos (one-hot encoding) de $L \times 4$, donde cada columna representa uno de los cuatro nucleótidos posibles (A, C, G, U) en cada posición. MiRe2e procesa esta entrada con tres modelos profundos internos, como se muestra en la Figura 2.2: Predicción de Estructura (A), Estimación de MEL (B) y Clasificador de pre-miARN (C). La figura muestra el modelo miRe2e completo, donde las entradas/salidas de cada submodelo se muestran con números y los detalles de cada arquitectura neuronal se muestran inmediatamente a continuación. El primer modelo permite obtener la estructura secundaria a partir de una secuencia de entrada de ARN. El modelo de estimación de la mínima energía libre del plegado (MEL) estima esta cantidad a partir de una secuencia de ARN de entrada y su estructura secundaria correspondiente. Finalmente, el último modelo profundo realiza la clasificación de la secuencia de entrada.

2.2.1. Predicción automática de estructuras de ARN

El modelo de predicción de estructura (Figura 2.2A) aprende a estimar la estructura secundaria a partir de una secuencia de ARN. Aquí, el tensor 1 entra en una red neuronal convolucional de tres etapas, cada una con bloques de identidad. Cada uno de estos bloques de identidad se compone de dos funciones de activación, dos capas de normalización por lotes, dos capas convolucionales unidimensionales de largo L y w_A filtros con conexiones de identidad (He *et al.*, 2016). La función principal de esta parte del modelo es extraer automáticamente motivos de la secuencia de entrada y aumentar el número de características para permitir un procesamiento rápido en las siguientes capas de atención (Vaswani *et al.*, 2017). En la salida de la CNN se agrega la señal de codificación de posición a cada embedding (Vaswani *et al.*, 2017). Luego, hay una pila de seis codificadores de tipo Transformer (Devlin *et al.*, 2018b). En esta parte del modelo, cada capa de encoder se compone de w_A características de entrada, h_A cabezales y n_A neuronas en las capas ocultas de cada red con propagación hacia adelante, donde el número de neuronas ocultas se establece en $n_A = 4w_A$ como se sugiere en (Vaswani *et al.*, 2017). La función de este encoder es, a través de sus mecanismos de atención, modelar la matriz de contacto de cada nucleótido en la secuencia de entrada, pudiendo así estimar su estructura secundaria. Finalmente, después del encoder hay un perceptrón multicapa de 3 capas, en donde se utilizan funciones de activación unidad lineal exponencial (ULE) en las capas ocultas y funciones tangentes hiperbólicas en la salida.

2.2.2. Estimación de la energía libre mínima

El segundo modelo (Figura 2.2B) tiene como objetivo estimar el MEL a partir de la secuencia de entrada y su estructura secundaria. Recibe las entradas 1 y 2, las concatena y obtiene un tensor $5 \times L$ siendo la quinta fila la estructura secundaria predicha para la secuencia de entrada. El modelo se compone de una CNN de 3 etapas, cada una compuesta por un bloque de identidad y una capa de pooling apilada. Debido al pooling, después de cada etapa la longitud del tensor de entrada se reduce en un factor de 2. En cada bloque de identidad las capas convolucionales unidimensionales están formadas por w_B filtros y poseen una dimensión de salida $w_B \times L/(2^C)$, donde C es el número de etapa. Finalmente, luego de aplanar el tensor hay un perceptrón multicapa de 3 capas donde cada una de estas tiene funciones de normalización por lotes y activación ULE. Para el entrenamiento se utilizó el error cuadrático medio, como la función de error entre cada valor de salida predicho y su valor de MEL de referencia. La salida de esta CNN es 3, el MEL estimado para la secuencia.

2.2.3. Modelo para clasificación de pre-miARN

El modelo clasificador de pre-miARN (Figura 2.2C) clasifica la secuencia de entrada 1, con su estructura secundaria 2 y el MEL estimado 3. Este modelo tiene una CNN de 4 etapas, cada una compuesta por tres bloques de identidad con w_C filtros y una capa de pooling apilada. Luego, hay una pila de tres codificadores de tipo Transformers. Cada capa de codificador tiene w_C características de entrada, h_C cabezas y n_C neuronas en las capas ocultas de cada red de propagación hacia adelante. Su función es codificar la información secuencial de la entrada, modelando así la dependencia entre cada nucleótido de forma global. Después del codificador, el tensor de salida con dimensión $w_C \times L/16$ se

aplana y concatena con la salida del modelo MEL [3]. Esto pasa a un perceptrón multicapa de 4 capas, funciones de activación de ULE, normalización de lotes y drop-out. Finalmente, una capa softmax en la salida predice la clase correspondiente [4] para la secuencia de entrada. Dado que miRe2e está compuesto por tres modelos en cascada, se realizó un entrenamiento de 3 etapas, donde la salida de cada modelo fue la entrada del siguiente.

Sección 3

Resultados

3.1. Medidas de complejidad del miARN maduro

3.1.1. Datos

Para el estudio de las nuevas medidas de complejidad se crearon conjuntos de datos con diferentes desbalance de clases, probando predictores de pre-miARN con y sin las medidas propuestas. Para esto hemos utilizado un conjunto de datos públicos disponible (Gudyś *et al.*, 2013), que proporciona ejemplos negativos y positivos de todos los pre-miRNA conocidos en miRBase para *Homo sapiens* (1.406 positivos y 81.228 negativos). Las características estándar son las utilizadas en los trabajos más citados (Stegmayer *et al.*, 2018; Jiang *et al.*, 2007; Gudyś *et al.*, 2013; Batuwita and Palade, 2009). Las proporciones variables de casos de cada clase permite evaluar la robustez de las nuevas características en situaciones más cercanas a las encontradas en un genoma real, donde el número de miARNs positivos conocidos es muy bajo con respecto al número de horquillas sin miARN en el resto de un genoma completo. Con este fin, se generaron conjuntos de datos mediante un muestreo aleatorio desde 1:500 (1 positivo en 500 negativos) hasta un desbalance extremo de 1:10.000.

3.1.2. Medidas de desempeño

Para reportar los resultados se utilizaron las medidas de

$$\text{Sensibilidad } s^+ = \frac{TP}{TP+FN}, \qquad \text{Precisión } p = \frac{TP}{TP+FP},$$

$$\text{Especificidad } s^- = \frac{TN}{TN+FP}, \qquad \text{F-score } F_1 = 2\,\frac{s^+\,p}{p+s^-},$$

Coeficiente de correlación de Matthew

$$CCM = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

Coeficiente Kappa

$$\kappa = \frac{a - a_c}{1 - a_c},$$

Donde TP, TN, FP y FN son el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, respectivamente. T es el número total de observaciones; $a = (TP + TN)/T$ es el número de aciertos estándar y a_c es el número de aciertos por azar, es decir, el proporcionado por un clasificador que asigna aleatoriamente una etiqueta positiva o negativa a cada muestra.

3.1.3. Configuración experimental

Para calcular las características se predijo la estructura secundaria de todas las secuencias (positivas y negativas) con RNAfold (Lorenz *et al.*, 2011), con 37°C y el resto de parámetros por omisión. Después de eso, se extrajeron las cadenas 5p y 3p cortando en 40 nt a partir del bucle terminal. De esta forma no se requiere la posición específica del miARN maduro dentro de la cadena y es posible calcular las características sin ninguna información adicional para horquillas desconocidas. Esto es importante porque se pueden generar diferentes miARN maduros a partir de la misma cadena (isomiRs) dependiendo de la posición del corte (Bartel, 2018).

El rendimiento en cada experimento se informa como el valor promedio de 8 particiones para los desbalances de 1:500 a 1:1.000, y 4 particiones para los desbalances de 1: 1.500 a 1:10.000, siempre usando solo la partición de prueba para medir los desempeños. La diferencia en el número de particiones seleccionadas para cada caso se debe a la disminución del número de positivos cuando aumenta el desbalance. Para evaluar si existe una diferencia estadísticamente significativa en el desempeño de los conjuntos de características propuestos, se realizó la prueba de Friedman para la medida F_1 con un nivel de significancia de $\alpha = 0{,}01$. Finalmente, para evaluar qué características tienen desempeños estadísticamente diferentes, se utilizó la prueba post-hoc de Nemenyi (Demšar, 2006).

La DL debe calcularse teniendo en cuenta que el conjunto de referencia (los pre-miARN positivos) cambia con cada partición de entrenamiento. Por lo tanto, solo se utilizan los miARN maduros que se encuentran en cada conjunto de entrenamiento $\mathcal{A}$ de cada partición de entrenamiento correspondiente, evitando así introducir información a priori del respectivo conjunto de prueba. Para las secuencias de entrenamiento la distancia de cada muestra de entrenamiento $\boldsymbol{a}_\ell \in \mathcal{A}$ se calcula como $L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \max\{L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p}), L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})\}$. En el caso de las muestras de prueba $\boldsymbol{t}_\ell$, todas las secuencias del conjunto de entrenamiento pueden ser usadas y la característica se calcula como $L_{\mathcal{A}}(\boldsymbol{t}_\ell) = \max\{L_{\mathcal{A}}(\boldsymbol{t}_\ell^{5p}), L_{\mathcal{A}}(\boldsymbol{t}_\ell^{3p})\}$.

Para el cálculo de EP se seleccionó $N = 2$ porque este valor mostró el mejor desempeño en las pruebas preliminares. Codificamos cada nucleótido A, C, G, U con un número entero de 1 a 4 según sus frecuencias relativas en las secuencias. Para combinar la información de ambas cadenas 3p y 5p, calculamos EP para cada una y seleccionamos la más pequeña. Es decir, el EP de orden 2 de cada candidato de prueba $\boldsymbol{t}$ se calculó como $EP_2(\boldsymbol{t}) = \min\{EP_2(\boldsymbol{t}^{5p}), EP_2(\boldsymbol{t}^{3p})\}$. De la misma manera, la LZ de cada candidato de prueba $\boldsymbol{t}$ se calculó como $LZ(\boldsymbol{t}) = \min\{LZ(\boldsymbol{t}^{5p}), LZ(\boldsymbol{t}^{3p})\}$.

3.1.4. Resultados de clasificación

Las Tablas 3.1 a 3.4 presentan los resultados para cada nueva característica y las características estándar (CE), para los clasificadores NB, RF, KNN y DBN, respectivamente. En cada fila, se informa el desempeño de cada clasificador para un desbalance dado, para todas las características, de acuerdo con CCM, κ y F_1. El mejor rendimiento para cada desbalance y cada medida se muestra en negrita.

La Tabla 3.1 muestra que, para NB con la DL versus las CE, las medidas de desempeño reflejan mejoras consistentes para todos los desbalances. En particular, cuando se utiliza la DL este clasificador obtuvo los mejores desempeños para todos los desbalances. Para el caso en el que se utiliza la EP, se encuentran mejoras con respecto a las CE para todas las medidas excepto para los desbalances de 1:2.000 y 1:4.000, donde el desempeño sigue siendo el mismo. En el caso de la LZ se observa el mismo comportamiento que para EP.

En la Tabla 3.2, al analizar el rendimiento de RF con las nuevas características, las tres medidas de desempeño muestran resultados consistentes, es decir, mejoran el rendimiento del clasificador en relación a CE solo. Desde 1:8.000 en adelante todas las medidas muestran que este clasificador se ve muy afectado por el desbalance. Del análisis de esta tabla de forma general, se puede observar que los mejores resultados para cada desbalance se distribuyen entre las tres características, pero siempre superando la CE en todos los casos y medidas.

La Tabla 3.3 muestra el desempeño del clasificador KNN con la DL versus las CE. Se puede ver aquí, nuevamente, que hay una mejora en el desempeño al incorporar la DL para desbalances menores a 1:8.000. La única excepción es para el desbalance de 1:4.000, donde solo el F_1 muestra una mejora en el desempeño del clasificador, mientras que las otras medidas muestran el mismo resultado que las CE solas. Las otras dos características mejoran las CE, pero solo ligeramente y para algunos casos. En

Tasa de	CE			CE+DL			CE+EP			CE+LZ		
desbalance	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1
1:500	0,314	0,197	0,200	**0,324**	**0,207**	**0,210**	0,315	0,198	0,201	0,317	0,199	0,202
1:1.000	0,223	0,107	0,111	**0,234**	**0,115**	**0,119**	0,227	0,109	0,113	0,224	0,108	0,111
1:2.000	0,180	0,066	0,067	**0,184**	**0,069**	**0,071**	0,179	0,065	0,067	0,179	0,065	0,067
1:4.000	0,166	0,056	0,058	**0,180**	**0,066**	**0,067**	0,166	0,056	0,058	0,167	0,057	0,058
1:6.000	0,142	0,040	0,044	**0,164**	**0,052**	**0,057**	0,146	0,042	0,046	0,143	0,040	0,044
1:8.000	0,143	0,040	0,041	**0,178**	**0,061**	**0,063**	0,145	0,041	0,043	0,146	0,042	0,044
1:10.000	0,130	0,038	0,041	**0,153**	**0,052**	**0,061**	0,134	0,040	0,043	0,134	0,040	0,042

Tabla 3.1: Resultados de clasificación de Naive Bayes para las características estándar (CE), distancia de Levenshtein (DL), entropía de permutación (EP) y Lempel-Ziv (LZ). Resultados reportados con coeficiente de correlación de Matthew (CCM), coeficiente Kappa (κ) y F_1 score.

Tasa de	CE			CE+DL			CE+EP			CE+LZ		
desbalance	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1
1:500	0,650	0,630	0,633	0,664	0,646	0,646	0,664	0,646	0,646	**0,682**	**0,666**	**0,654**
1:1.000	0,602	0,532	0,510	**0,612**	**0,545**	**0,526**	0,498	0,456	0,453	0,591	0,518	0,492
1:2.000	0,418	0,298	0,279	**0,500**	**0,400**	0,372	0,447	0,333	0,311	**0,500**	**0,400**	**0,380**
1:4.000	0,447	0,333	0,266	0,387	0,261	0,208	**0,500**	**0,400**	**0,339**	0,387	0,261	0,194
1:6.000	-1,000	0,000	0,000	**0,289**	**0,154**	**0,125**	-1,000	0,000	0,000	-1,000	0,000	0,000
1:8.000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000
1:10.000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000

Tabla 3.2: Resultados de clasificación de Random Forest para las características estándar (CE), distancia de Levenshtein (DL), entropía de permutación (EP) y Lempel-Ziv (LZ). Resultados reportados con coeficiente de correlación de Matthew (CCM), coeficiente Kappa (κ) y F_1 score.

Tasa de	CE			CE+DL			CE+EP			CE+LZ		
desbalance	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1
1:500	0,531	0,530	0,527	**0,568**	**0,568**	**0,574**	0,531	0,530	0,531	0,531	0,530	0,531
1:1.000	0,421	0,421	0,411	**0,441**	**0,441**	**0,447**	0,421	0,421	0,414	0,409	0,409	0,419
1:2.000	0,399	0,373	0,383	**0,494**	**0,476**	**0,478**	0,372	0,345	0,356	0,448	0,426	0,414
1:4.000	**0,592**	**0,518**	0,451	**0,592**	**0,518**	**0,476**	0,404	0,400	0,442	0,592	**0,518**	0,451
1:6.000	0,408	0,286	0,250	**0,577**	**0,500**	**0,367**	0,408	0,286	0,225	0,408	0,286	0,225
1:8.000	0,354	0,222	0,167	0,354	0,222	0,167	0,354	0,222	0,167	0,354	0,222	0,167
1:10.000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000	-1,000	0,000	0,000

Tabla 3.3: Resultados de clasificación de k-nearest neighbor para las características estándar (CE), distancia de Levenshtein (DL), entropía de permutación (EP) y Lempel-Ziv (LZ). Resultados reportados con coeficiente de correlación de Matthew (CCM), coeficiente Kappa (κ) y F_1 score.

el punto de desbalance más alto, KNN tiene un rendimiento extremadamente bajo, el cual se refleja en todas las medidas.

En la Tabla 3.4, al analizar el rendimiento de DNN con DL frente a las CE, se observa una mejora significativa en las tres medidas de rendimiento, y para todos los desbalances, cuando se agrega la nueva DL a las CE. Para el caso de la EP en relación a las CE, se observa que CCM y κ muestran mejoras para los desbalances mayores de 1:6.000. Con F_1 se encuentra la misma mejora para todos los casos.

Finalmente, después de un análisis exhaustivo de las cuatro tablas de esta sección, se puede afirmar que con las características propuestas se observan mejoras para todas las medidas de desempeño, de manera consistente e independiente del clasificador utilizado. Se puede observar que RF y KNN muestran valores iguales a cero (o CCM de -1.0) para los desbalances más grandes. Esto se debe al sesgo generado por las probabilidades a priori de las clases, que hace que el clasificador etiquete todos los

Tasa de	CE			CE+DL			CE+EP			CE+LZ		
desbalance	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1	CCM	κ	F_1
1:500	0,704	0,702	0,695	**0,725**	**0,724**	**0,714**	0,697	0,697	0,707	0,704	0,702	0,693
1:1.000	0,499	0,492	0,488	**0,544**	**0,508**	**0,493**	0,472	0,451	0,453	0,483	0,464	0,461
1:2.000	0,508	0,506	0,496	**0,617**	**0,617**	**0,622**	0,506	0,490	0,548	0,495	0,494	0,483
1:4.000	0,564	0,564	0,603	**0,699**	**0,698**	**0,708**	0,600	0,600	0,611	0,699	**0,698**	0,648
1:6.000	0,400	0,400	0,293	**0,764**	**0,737**	**0,579**	0,333	0,333	0,268	0,463	0,461	0,381
1:8.000	0,320	0,316	0,325	**0,935**	**0,933**	**0,775**	0,408	0,400	0,274	0,408	0,400	0,325
1:10.000	0,320	0,316	0,278	**0,866**	**0,857**	**0,783**	0,612	0,545	0,392	0,612	0,545	0,433

Tabla 3.4: Resultados de clasificación de Deep Belief Networks para las características estándar (CE), distancia de Levenshtein (DL), entropía de permutación (EP) y Lempel-Ziv (LZ). Resultados reportados con coeficiente de correlación de Matthew (CCM), coeficiente Kappa (κ) y F_1 score.

casos positivos como parte de la clase mayoritaria (clase negativa). También se observa que DNN logró los mejores desempeños para todos los desbalances y todas las características propuestas, demostrando además que estas mejoras se reflejan igualmente en las tres medidas de desempeño reportadas. Por este motivo, en el resto de esta sección, sólo se utilizará este clasificador para el análisis detallado del comportamiento de las características propuestas. Además, debido a que las tres medidas reportan un comportamiento similar, a partir de ahora se utilizará solamente F_1.

3.1.5. Desempeño detallado de las nuevas características

La Figura 3.1 muestra un análisis detallado de los resultados de la clasificación para cada una de las nuevas características propuestas y las CE, con una DBN como clasificador. El eje horizontal muestra el nivel de desbalance entre clases, mientras que el eje vertical muestra la sensibilidad s^+, precisión p y F_1, en las Figuras 3.1a, 3.1b y 3.1c, respectivamente.

La Figura 3.1 muestra claramente cómo el clasificador DBN es capaz de mantener un buen desempeño cuando se incrementa el desbalance, e incluso aumentar tanto s^+ (Figura 3.1a) como p (Figura 3.1b) cuando se utiliza la nueva característica DL. Este es un resultado notable, que tiene un impacto directo en el buen desempeño de DBN con DL en F_1. En la Figura 3.1c, al analizar el desempeño de DBN con CE respecto a la DL, se observa que F_1 es significativamente mayor para todos los desbalances cuando se utiliza la nueva característica DL. Por ejemplo, se puede ver que para los desbalances entre 1:500 y 1:10.000, F_1 con CE desciende de casi el 70 % a alrededor del 20 %. En este mismo rango de desbalance, sin embargo, DBN con la DL sube hasta casi el 80 %. También se puede notar que la precisión del clasificador aumenta mucho con la incorporación de la DL hasta un nivel muy alto (superior al 90 %) en el mayor desbalance aquí estudiado (1:10.000). Este es un resultado muy importante en términos prácticos, especialmente para los desbalances más cercanos a los casos reales en los que se utilizan datos de un genoma completo, porque asegura reducir notablemente la cantidad de falsos positivos. Debido al hecho que, en términos generales, s^+ también mejora cuando se utiliza la DL, la F_1 crece para todos los casos a medida que se incrementa el desbalance. Esto es muy interesante ya que la capacidad de evitar falsos positivos parece ser robusta al desbalance y al tamaño del conjunto positivo, sin por ello influir en la detección de casos positivos. Al analizar todas las figuras de manera global, se observa una mejora de la DL con respecto a las CE para todas las medidas, lo cual presenta una clara tendencia a crecer a medida que se incrementa el desbalance. Por otro lado, las otras características tienen un rendimiento más variable. En resumen, se puede afirmar que se obtiene una mejora muy importante en el desempeño cuando se utiliza la DL en el conjunto de características, incluso para el caso del desbalance más alto.

Para DBN con la EP se observa una mejora para F_1 de aproximadamente un 10 % solo para el mayor desbalance, en donde F_1 es casi 30 % con las CE y casi 40 % cuando también se usa la EP. La mejora más importante y notable se observa en p para 1:10.000, en donde dicha medida sube del 30 % a más del 45 %. Esto sugiere que esta característica puede reducir efectivamente los falsos positivos, logrando una mejora de la precisión para problemas en donde se encuentre un alto desbalance. En resumen, se puede afirmar que la EP solo puede mejorar el rendimiento de las DBN para los casos de

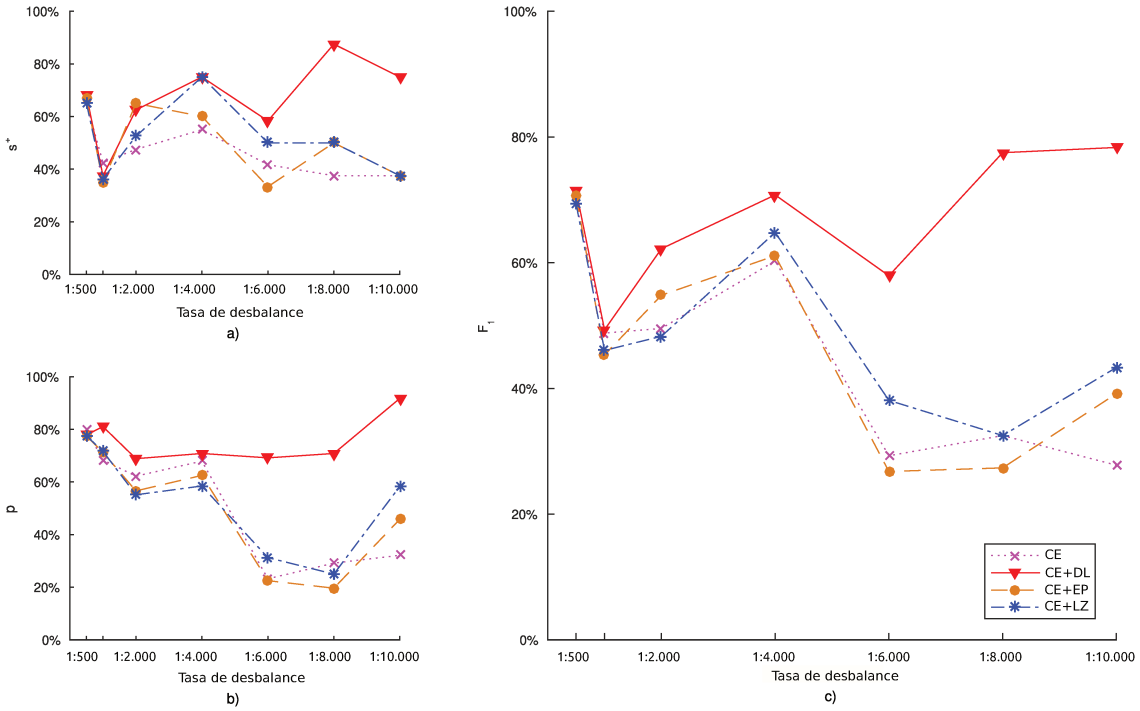

Figura 3.1: Resultados de Deep Belief Networks (DBN) con características estándar (CE), distancia de Levenshtein (DL), entropía de permutación (EP) y Lempel-Ziv (LZ). a) Sensibilidad, s^+; b) Precisión, p; c) F_1.

grandes desbalances.

En el caso de LZ, al analizar el desempeño de DBN con las CE versus la DBN con la incorporación de LZ, se observa que F_1 es superior para el desbalance más grande. También se puede ver que la mejora de F_1 se debe a una ligera mejora de p y s^+. Es decir, LZ probablemente puede servir para evitar falsos positivos, especialmente cuando la clase negativa es extremadamente grande con respecto a la clase positiva. Se puede afirmar, en resumen, que LZ puede tener la capacidad de mejorar el rendimiento de una DBN para altos desbalances, principalmente gracias a la mejora de p.

3.1.6. Desempeño global de las nuevas características

La Tabla 3.5 muestra los resultados de diferentes combinaciones de las características propuestas para el clasificador DBN. En cada fila se puede observar F_1 para los diferentes conjuntos de características y para cada desbalance. Se puede ver que la DL mejora el rendimiento del clasificador en todos los casos, incluso para los desbalances muy altos (1:10.000). En cambio, LZ y la EP individualmente no mejoran el rendimiento de la DBN. F_1 en esos casos permanece igual o bastante similar al caso las CE. Observando las diferentes combinaciones de características para la DBN, se puede notar que F_1 mejora para todos los casos en DL + EP con respecto a las CE. Además, para el caso de 1:2.000, 1:4.000 y 1:6.000, DL + EP combinadas logran un rendimiento mayor que cuando se usan por separado. Para DL + LZ, F_1 mejora para todos los casos con respecto a las CE (excepto 1:1.000, donde permanece casi igual). Además, para los casos de 1:4.000 y 1:8.000, DL + LZ supera el rendimiento de las características usadas por separado. En el caso de EP + LZ se observa que F_1 en su mayoría permanece igual, o mejora sólo ligeramente para algunos casos. Finalmente, analizando el comportamiento de la combinación de todas las características juntas, se puede afirmar que F_1 mejoró en todos los casos.

La Tabla 3.5 muestra, de manera más global, dos resultados clave y complementarios. En primer

Desbalance	CE	DL	EP	LZ	DL+EP	DL+LZ	EP+LZ	Todas
1:500	69,50	**71,44**	70,65	69,34	71,39	**71,68**	68,96	71,50
1:1.000	48,81	**49,33**	45,33	46,05	49,26	48,71	52,85	**53,85**
1:2.000	49,55	**62,22**	54,82	48,29	63,21	57,72	53,33	**65,34**
1:4.000	60,28	**70,78**	61,11	64,81	**78,28**	73,33	64,95	71,89
1:6.000	29,29	**57,92**	26,79	38,10	**61,67**	57,92	29,17	56,79
1:8.000	32,50	**77,50**	27,36	32,50	77,50	**85,00**	36,67	77,50
1:10.000	27,78	**78,33**	39,17	43,33	62,50	**70,00**	40,48	54,17

Tabla 3.5: Resultados de F_1 para diferentes combinaciones de la distancia de Levenshtein (DL), entropía de permutación (EP) y complejidad de Lempel-Ziv (LZ) con DBN. En negrita se presentan los mejores resultados para cada panel de la tabla, individual (izq) y combinado (der).

lugar, la DL es la característica que tiene el mejor desempeño individual. En segundo lugar, aunque las características EP y LZ mejoran individualmente los resultados del clasificador DBN, sus contribuciones tienen más impacto cuando se combinan. Por ello se puede decir que las nuevas características presentadas en esta tesis aportan información de forma complementaria entre ellas.

Para evaluar la significancia estadística de los resultados se realizó la prueba de Friedman para F_1, resultando un valor p de 2.5748E-05 ($\alpha = 0{,}01$), el cual indica que existe una diferencia estadísticamente significativa entre los resultados. Luego, se realizó la prueba post-hoc de Nemenyi para F_1. Esta prueba mostró que entre DL y DL + EP + LZ, el primer grupo, y LZ, la EP y las CE, el segundo grupo, hay una diferencia estadísticamente significativa. De esta manera, los resultados obtenidos indican que la DL y la combinación DL + EP + LZ son las mejores características, en comparación con las CE, LZ y la EP por separado. Sin embargo, no se encontró una diferencia estadísticamente significativa entre DL y DL + EP + LZ. Además, debido al hecho de que hay muy pocas muestras positivas en las particiones de prueba para los desbalances más altos, se repitió el experimento 10 veces con diferentes muestreos de los ejemplos positivos para el caso de la DL versus las CE con la DBN para el desbalance de 1:10.000. De esta manera, se obtuvo una mediana para F_1 de 66,67 % y 30,95 %, para la DL y las CE respectivamente. Se realizó una prueba de rango con signo de Wilcoxon a estas 40 particiones de prueba y se obtuvo una p <6.2028E-05.

3.1.7. Discusión

Un punto interesante para discutir es por qué la EP y LZ individualmente no han mostrado un comportamiento robusto al incrementarse el desbalance. Sin embargo, cuando se combinan con la DL, se encontró que estas realmente ayudan a mejorar la robustez al desbalance. Este comportamiento sugiere que la EP y la complejidad de LZ pueden capturar información útil de los miARN maduros, pero debido a su longitud pequeña no les es posible obtener valores suficientemente discriminativos, por sí mismas, separadamente. Sin embargo, estas son más discriminativas cuando se combinan con la DL, debido a que esta característica no depende de la longitud de la secuencia en sí misma, sino de la distancia al conjunto de referencia completo, como se explicó anteriormente. Por este motivo, cuando se combinan todas las características se observa un predominio de la DL sobre la EP y LZ, aunque la inclusión de estas últimas sigue aportando alguna información discriminativa. Por ejemplo, para un desbalance de 1:2.000, la línea de base F_1 proporcionada por las CE es 49,55 %, la DL la mejora hasta 62,22 % pero la EP y LZ son ligeramente mejores que las CE. Así, el 65,34 % de DL+EP+LZ está claramente dominado por la DL. Por otro lado, los mejores resultados de la característica de la distancia de Levenshtein se pueden explicar en base al hecho de que esta se calcula respecto a un conjunto externo de pre-miARNs. En cambio, la entropía de permutación y la complejidad de Lempel-Ziv se calculan sólo con información perteneciente a la propia secuencia. Es decir, la DL permite tener una medida más precisa y un sentido representativo de pertenencia a la clase positiva, dado que la Dl es una distancia a un conjunto de referencia de miARNs. Desde otro punto de vista, esto sugiere que el miARN maduro contiene ciertas estructuras sintácticas que guían su funcionamiento, evitando así cualquier mutación aleatoria que lo modifique. Por lo tanto, al combinar la información de la mediana de la distancia para un candidato (DL), junto con la información de su aleatoriedad (EP)

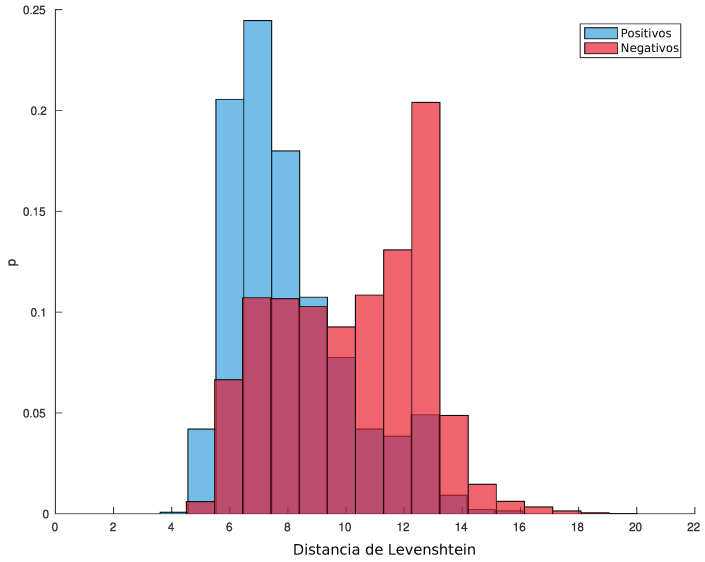

Figura 3.2: Histograma de la característica DL para las clases positiva y negativa, para el conjunto de datos completo (1.406 positivos y 81.228 negativos)

y su complejidad (LZ), estamos restringiendo el número de secuencias candidatas solo a las posibles combinaciones de nucleótidos que pueden permitir pequeños cambios, con una complejidad definida.

Por otro lado, quisiéramos entender por qué DL muestra un comportamiento tan robusto al desbalance. Generalmente, los algoritmos para la predicción de pre-miARN utilizan bases de datos públicas para el entrenamiento, lo que genera un sesgo hacia pre-miARNs previamente conocidos. Dado que la mayoría de ellos tienen una estructura de tallo-bucle, y la mayoría de las características se basan en esa estructura, con estas características estándar es difícil reconocer posibles nuevos pre-miARNs que difieran de los canónicos. Sin embargo, la inclusión de una característica de secuencia como la DL, calculada a partir del miARN maduro, es disruptiva en este sentido porque permite tener en cuenta información adicional de los candidatos, aún cuando la estructura secundaria sea bastante diferente a la canónica. Así, en un espacio diferente, generado por las características nuevas, las distancias cambian y las secuencias que no estaban cercanas según las características estándar ahora pueden estar próximas en el espacio nuevo generado con la información del miARN maduro. Una segunda explicación para el éxito de la DL es que no se calcula solo con la información de cada candidato, sino que es una distancia de cada secuencia con respecto a todo el conjunto de referencia. Y un tercer punto de vista es que se puede decir que esta característica podría ser capaz de obtener una gran robustez frente a secuencias candidatas que pueden tener una estructura más reciente desde el punto de vista evolutivo. Esto se debería a la incorporación de información proveniente del miARN maduro, la cual es complementaria a la estructura de cada candidato. Por todo esto, esta nueva característica podría permitir encontrar nuevos pre-miARN que difieran de los pre-miARN canónicos.

Un último tema importante para la discusión es si los resultados de la DL pueden estar sesgados hacia familias o clases de miARNs mayoritarias. En (2.1) la DL se calcula como un estadístico de las distancias a cada miARN maduro del conjunto de entrenamiento, pero la elección de este estadístico no fue trivial. En primer lugar, se eligió el mínimo para evitar un posible sesgo hacia las familias más numerosas. Sin embargo, los resultados obtenidos mostraron una amplia superposición de ambas clases, debido a que el mínimo considera solo la secuencia más similar. Por el contrario, la mediana mostró ser un estadístico más informativo debido a que tiene en cuenta el conjunto completo de entrenamiento. De esta manera, se mostró que se obtiene una distribución de clases con una mayor separación (ver Figura 3.2).

3.2. Predicción de pre-miARN con aprendizaje profundo

3.2.1. Datos

En estos experimentos se utilizaron datos de todo el genoma de *Homo sapiens*[1]. Para entrenar el primer modelo (predicción de estructura secundaria), se utilizaron todos los pre-miARNs de metazoos (23.178) obtenidos de mirBase v.22, excluyendo *H. sapiens*, y se extrajeron 2.000.000 de pseudo-horquillas del genoma con HextractoR (Yones *et al.*, 2020). La salida de referencia para cada secuencia de entrada es su correspondiente estructura secundaria predicha con RNAfold (Hofacker, 2003), a una temperatura de 37°C. Para entrenar el modelo profundo que estima la MEL se requiere la estructura secundaria predicha por el primer modelo y su respectiva secuencia de entrada de ARN. La salida deseada aquí fue el valor de MEL predicho por RNAfold normalizado por la longitud de la secuencia. Para este modelo se utilizaron 23.178 pre-miARNs de metazoos (excluyendo *H. sapiens*). Además, se extrajeron aleatoriamente del genoma 48.000 pseudo-horquillas obtenidas con HextractoR y 48.000 secuencias que no formaban horquillas (planas). Para la etapa de prueba del modelo completo las secuencias de entrada fueron obtenidas mediante un escaneo y corte de cada cromosoma con ventanas superpuestas (longitud 100 nt, paso 20 nt).

3.2.2. Medidas de desempeño

El desempeño de los métodos es reportado con las métricas de evaluación de sensibilidad (s^+), precisión (p) y F_1. Estas medidas además fueron usadas para obtener las curvas de precisión-sensibilidad (PRC), la cual es un buen indicador del desempeño global de un clasificador. Como se ha mostrado en (Saito *et al.*, 2015), esta medida se prefiere por sobre la clásica curva característica operativa del receptor (ROC) para evaluar un clasificador binario en datos altamente desbalanceados. El área debajo de la curva de precisión-sensibilidad (AUCPR), la cual es un simple resumen numérico de la información en la curva, será reportada como una medida global acerca de todos los posibles umbrales de salida computados en el modelo.

3.2.3. Comparación con el estado del arte

Para mostrar mejor la capacidad de generalización de nuestro modelo, se realizó una comparación de predicciones con validación cruzada para el cromosoma 1 de *H. sapiens*. Los datos de entrenamiento incluyeron todos los positivos (156 pre-miARN conocidos) en el cromosoma 1 y el resto de las secuencias del cromosoma 1 (más de 24.000.000), divididos en 4 particiones para entrenamiento y prueba. Comparamos el rendimiento obtenido con miRe2e para esta tarea con el modelo de predicción de pre-miARNs propuesto más recientemente, deepMir (Tang and Sun, 2019), que también recibe secuencias de entrada sin procesar.

Los resultados se muestran en la Tabla 3.6, que informa los resultados de cada partición en las filas, y s^+, p y F_1 en las columnas para cada método. En cuanto a s^+, ambos métodos tienen buenos resultados, siendo deepMir ligeramente mejor en promedio. En cambio, la precisión de miRe2e es siempre superior, para todas las particiones . Es muy notable aquí que el rendimiento de miRe2e es de un orden de magnitud más alto que el de deepMir. Esto se refleja precisamente en F_1, donde miRe2e es siempre superior a deepMir, en todos los casos con un orden de magnitud de diferencia. Esto se debe al hecho de que miRe2e puede modelar eficazmente la estructura secundaria de la secuencia de ARN, lo que es clave para filtrar falsos positivos. De este modo, el modelo puede mejorar p sin una caída en s^+, aumentando así el F_1 global. Cabe señalar que estos resultados se obtuvieron en el contexto del alto desbalance de un cromosoma (156 muestras positivas frente a 24.000.000 de muestras negativas), lo que sugiere que el rendimiento de miRe2e en un escenario de genoma completo seguirá siendo superior a deepMir.

[1] http://ftp.ensembl.org/

Partición	s^+		p		F_1	
	miRe2e	deepMir	miRe2e	deepMir	miRe2e	deepMir
1	0,0130	**0,0250**	**0,0020**	0,0002	**0,0030**	0,0005
2	**0,0130**	**0,0130**	**0,0020**	0,0004	**0,0040**	0,0008
3	**0,0380**	0,0130	**0,0010**	0,0006	**0,0020**	0,0012
4	0,0130	**0,1150**	**0,0110**	0,0005	**0,0120**	0,0011
Promedio	0,0193	**0,0415**	**0,0040**	0,0004	**0,0052**	0,0009

Tabla 3.6: Comparación de desempeño para miRe2e y deepMir en validación cruzada para la predicción de pre-miARNs en el cromosoma 1 de *H. sapiens*.

3.2.4. Predicción de pre-miARNs humanos agregados a miRBase en el futuro

Para probar el rendimiento de miRe2e en un escenario aún más realista, el cual implica la predicción de nuevos pre-miARN en el futuro, entrenamos el modelo con un conjunto de datos de pre-miARNs humanos de miRBase v17 (2011) y luego lo probamos con los pre-miARNs humanos introducidos posteriormente en miRBase v21 (2014). El conjunto de entrenamiento tuvo 1.854 secuencias positivas, 87.500 negativas y 787.500 planas, estando el conjunto de prueba compuesto por 27 secuencias positivas, 12.500 negativas y 112.500 planas. Las curvas PR se muestran en la Figura 3.3 para ambos modelos en diferentes colores. Se puede ver que miRe2e (línea azul) alcanzó los mejores resultados, con un AUCPR= $0,17$. El método deepMir (línea naranja) obtuvo un AUCPR= $0,08$, menos de la mitad que miRe2e. Cabe señalar que para la misma sensibilidad en ambos métodos (por ejemplo, $s = 0{,}20$), mientras que miRe2e obtiene una $F_1 = 0,26$ con 11 FP, deepMir tiene una $F_1 = 0,08$ con 113 FP (más de 10 veces). Esto es de gran importancia en el dominio de aplicación, donde sí para la misma tasa de TP se obtiene un gran número de nuevos candidatos a pre-miARN, del orden de cientos o miles, será casi imposible validarlos experimentalmente en el laboratorio biológico para descubrir cuáles de ellos son pre-miARNs reales. Por lo tanto, siempre es preferible que la predicción contenga un número menor de buenos candidatos, es decir una mejor precisión dado el alto desbalance. Estos resultados muestran que miRe2e es eficaz para la predicción y el descubrimiento de futuros pre-miARN.

3.2.5. Descubrimiento de pre-miARNs en el genoma completo de una nueva especie

Finalmente, se probó miRe2e en la tarea más realista de descubrir nuevos pre-miARN en una nueva especie. El modelo de predicción de estructura se entrenó con todos los pre-miARNs de metazoos conocidos, excluyendo *H. sapiens* (23.048 secuencias) y muestras negativas de animales (2.000.000 de horquillas en total de *Anopheles gambiae*, *Drosophila melanogaster* y *Caenorhabditis elegans*) (Bugnon *et al.*, 2020a). El modelo para la estimación de la MEL se entrenó con todos los pre-miARNs de metazoos conocidos excluyendo *H. sapiens*, 48.000 pseudo-horquillas y 48.000 secuencias planas extraídas al azar de *Anopheles gambiae*, *Drosophila melanogaster* y *Caenorhabditis elegans*. El modelo clasificador de pre-miARNs se entrenó con todos los pre-miARNs de metazoos conocidos, excluyendo *H. sapiens* (23.048), y muestras negativas de animales (1.000.000 en total de *Anopheles gambiae*, *Drosophila melanogaster* y *Caenorhabditis elegans*: 100.000 horquillas y 900.000 planas). La tarea consistía en el descubrimiento de todos los pre-miARNs del genoma humano, como si se tratara de una nueva especie descubierta recientemente. Por lo tanto, para la prueba todas las secuencias dentro de cada cromosoma que contienen un pre-miARN conocido, de acuerdo con las posiciones descritas en miRBase v22, se usaron como clase positiva, y las negativas fueron todas las secuencias correspondientes al resto del cromosoma.

Los resultados se presentan en la Tabla 3.7. La primera columna indica el cromosoma y la segunda y tercera columna el número de ejemplos positivos y negativos en ese cromosoma, respectivamente. Luego, el rendimiento de cada método se informa con s^+, p, F_1, AUROC y AUCPR. Finalmente, la última fila indica el desempeño final medido en el genoma humano completo. Cabe destacar el alto desbalance de clases que existe en cada cromosoma. Por ejemplo, en el cromosoma 1 hay 156

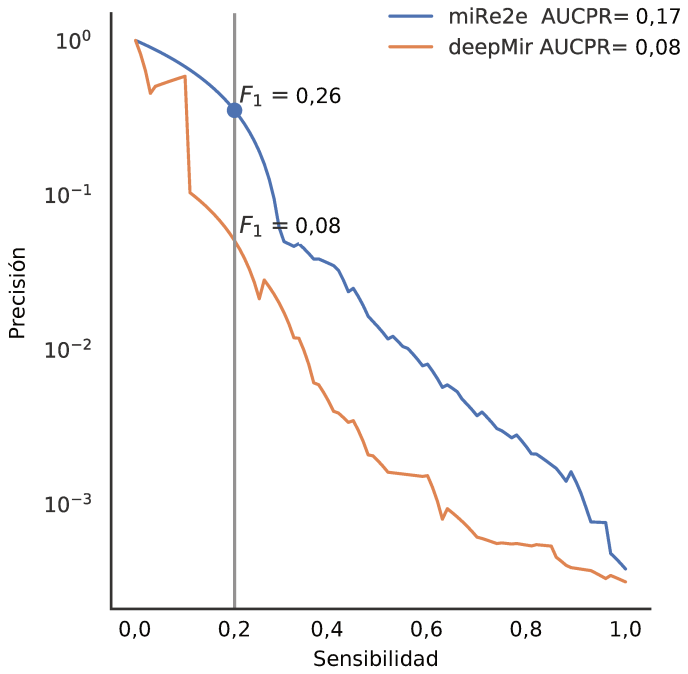

Figura 3.3: Curvas precisión-sensibilidad para la predicción de pre-miARNs humanos recientemente agregados a miRBase.

positivos en más de 24 millones de negativos, es decir, una relación de desbalance entre clases de aproximadamente 1:160.000. Peor aún, en el cromosoma Y solo hay 4 positivos y más de 5 millones de negativos, lo que hace que la relación de desbalance sea de 1:1.430.000. Debido a este alto nivel de desbalance en los cromosomas, no se pueden esperar valores muy altos de F_1. Nótese que en este caso, con solo el 1 % de FP, el F_1 cae por debajo de 0,0001, por lo que las medidas globales de AUROC y AUCPR son un complemento importante para el análisis de estos resultados.

Los resultados mostrados en la Tabla 3.7 indican que, a pesar del alto desbalance de clases existente en cada cromosoma, el modelo miRe2e tiene resultados competitivos y los mejores para todos los casos. Con respecto a la s^+, miRe2e es dos veces mejor que deepMir para todos los cromosomas. La precisión es mejor también, incluso un orden de magnitud mayor en la mayoría de los casos. En particular, para el cromosoma 2 el rendimiento de miRe2e en precisión es 20 veces mejor que el de deepMir. En el único caso en el que deepMir tiene llamativamente una $p = 1,00$ (cromosoma 5), debe notarse sin embargo que la sensibilidad correspondiente es $s^+ = 0,013$ (en contraste con $s^+ = 0,280$ para miRe2e). Aunque en este punto (s^+, p) deepMir maximiza F_1, esto se logra a costa de una sensibilidad muy baja. Para las medidas F_1 y AUROC, nuevamente, miRe2e supera claramente a deepMir en todos los cromosomas. Finalmente, con respecto a la mejor medida de desempeño para este tipo de problemas con alto desbalance de clases, AUCPR, puede verse fácilmente en la tabla que todos los mejores resultados corresponden a miRe2e.

Como comparación final, no solo con los métodos de aprendizaje profundo que utilizan datos sin procesar, sino también con uno de los mejores métodos actuales que utiliza la estructura secundaria predicha de las secuencias, hemos realizado un nuevo experimento en todo el genoma. La Figura 3.4 muestra las curvas PR para el genoma humano completo (usando todas las secuencias de todos los cromosomas), para miRe2e (datos sin pre-procesamiento), deepMir (datos sin pre-procesamiento) y deepMiRGene, que recibe tanto las secuencias como sus estructuras secundarias. Aunque el último no es un modelo profundo completo de extremo a extremo, debido a que utiliza la estructura secundaria predicha por un modelo externo no neuronal (RNAfold), proporciona una referencia de comparación válida con el mejor método del estado del arte. En la parte superior izquierda de la Figura 3.4 se puede ver claramente que el mejor desempeño es para miRe2e, con la mayor diferencia respecto a los otros

Crm	Pos	Negativos	deepMir					miRe2e				
			s^+	p	F_1	AUROC	AUPRC	s^+	p	F_1	AUROC	AUPRC
1	156	24.895.488	0,013	0,0020	0,0035	0,7115	0,00004	0,235	0,0040	0,0079	0,9439	**0,11880**
2	116	24.213.504	0,075	0,0002	0,0003	0,7081	0,00003	0,271	0,0038	0,0075	0,9640	**0,13673**
3	96	19.826.688	0,063	0,0001	0,0002	0,7024	0,00002	0,240	0,0043	0,0085	0,9623	**0,12117**
4	62	19.015.680	0,172	0,0000	0,0001	0,7442	0,00002	0,190	0,0025	0,0050	0,9724	**0,09576**
5	76	18.149.376	0,013	1,0000	0,0263	0,6978	0,01335	0,280	0,0045	0,0089	0,9585	**0,14131**
6	71	17.080.320	0,071	0,0001	0,0002	0,7885	0,00002	0,271	0,0043	0,0084	0,9771	**0,13684**
7	82	15.931.392	0,013	0,0004	0,0008	0,7880	0,00004	0,138	0,0019	0,0038	0,9537	**0,06949**
8	90	14.512.128	0,012	0,0030	0,0048	0,7488	0,00016	0,232	0,0044	0,0086	0,9196	**0,11732**
9	88	13.836.288	0,012	0,0014	0,0025	0,6767	0,00003	0,318	0,0056	0,0109	0,9580	**0,16041**
10	69	13.375.488	0,030	0,0001	0,0003	0,7041	0,00003	0,333	0,0044	0,0086	0,9676	**0,16804**
11	102	13.504.512	0,040	0,0004	0,0007	0,8100	0,00008	0,228	0,0042	0,0082	0,9669	**0,11528**
12	80	13.326.336	0,013	0,0526	0,0206	0,7694	0,01288	0,231	0,0042	0,0082	0,9382	**0,11621**
13	40	11.433.984	0,025	0,0000	0,0001	0,7227	0,00001	0,150	0,0027	0,0053	0,9581	**0,07596**
14	99	10.702.848	0,041	0,0004	0,0009	0,7364	0,00004	0,204	0,0061	0,0119	0,9726	**0,10385**
15	71	10.199.040	0,044	0,0002	0,0005	0,6519	0,00003	0,324	0,0066	0,0129	0,9564	**0,16360**
16	82	9.031.680	0,148	0,0003	0,0007	0,6709	0,00009	0,210	0,0035	0,0069	0,9427	**0,10615**
17	110	8.325.120	0,075	0,0002	0,0004	0,6709	0,00004	0,142	0,0028	0,0054	0,9501	**0,07162**
18	35	8.036.352	0,031	0,0001	0,0003	0,6778	0,00002	0,250	0,0040	0,0080	0,9430	**0,12595**
19	143	5.861.376	0,007	0,0014	0,0024	0,8321	0,00018	0,300	0,0077	0,0150	0,9661	**0,15277**
20	48	6.438.912	0,021	0,0024	0,0043	0,7646	0,02131	0,234	0,0034	0,0067	0,9727	**0,11774**
21	33	4.669.440	0,032	0,0067	0,0111	0,7025	0,03229	0,161	0,0034	0,0067	0,9610	**0,08132**
22	46	5.861.376	0,068	0,0002	0,0003	0,6272	0,00002	0,295	0,0038	0,0075	0,9412	**0,14882**
X	118	15.599.616	0,009	0,0013	0,0023	0,7605	0,00003	0,301	0,0096	0,0186	0,9741	**0,15368**
Y	4	5.720.064	0,250	0,0001	0,0003	0,5668	0,00002	0,500	0,0003	0,0006	0,7954	**0,00010**
Todos	1.917	309.547.008	0,004	0,0003	0,0006	0,7117	0,00003	0,244	0,0043	0,0085	0,9595	**0,12313**

Tabla 3.7: Comparación de desempeños para la predicción de pre-miARNs en el genoma de *H. sapiens*. Se detallan las medidas para cada cromosoma (crm) y el genoma completo (fila Todos)

métodos. Para sensibilidades más altas ($s^+ > 0,6$), miRe2e se comporta igual que deepMiRGene y mucho mejor que deepMir. Sin embargo, esta parte de la curva PR tiene una utilidad práctica muy limitada, debido al gran número de falsos positivos generados en este escenario altamente desbalanceado. Cabe mencionar que este rendimiento para miRe2e se obtiene sin requerir más información que la secuencia de ARN sin procesar. Sorprendentemente, en este experimento el AUCPR total para miRe2e es de 0,12313, más de 100 veces mayor al mejor de los otros métodos.

Estos resultados indican que miRe2e puede usarse de manera confiable para el descubrimiento de nuevos pre-miARNs en un genoma completo, con la mejor sensibilidad y precisión en este escenario de alto desbalance. Es decir, utilizando un número muy bajo de ejemplos positivos en el aprendizaje para el descubrimiento de nuevos candidatos. Esto convierte a miRe2e en el primer modelo completo de aprendizaje profundo de extremo a extremo, basado en Transformers, con resultados competitivos para la tarea de predicción de pre-miARN en genomas completos.

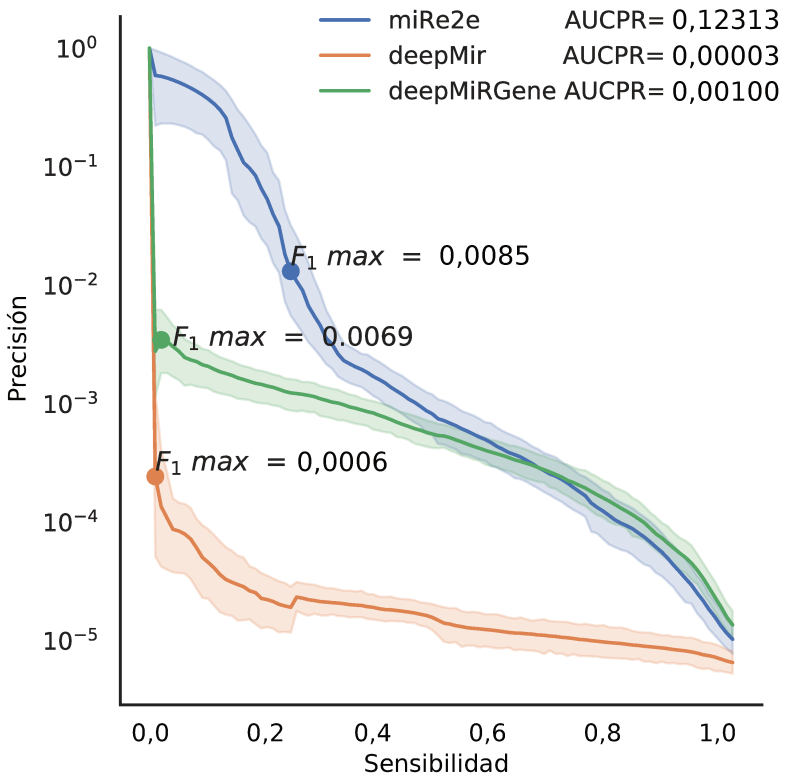

Figura 3.4: Curvas precisión-sensibilidad para miRe2e, deepMir y deepMiRGene para la predicción de pre-miARNs humanos en el genoma completo.

Sección 4

Conclusiones

En esta tesis doctoral se desarrollaron tres nuevas características basadas en medidas de complejidad para la predicción de pre-miARN. La principal motivación fue la de extraer información codificada a nivel secuencial en el miARN maduro. Los resultados mostraron que la incorporación de las medidas propuestas del miARN maduro proporcionan un alto poder discriminativo. Especialmente, la distancia de Levenshtein propuesta ha demostrado tener el mejor rendimiento para todos los desbalances. Además, las características propuestas basadas en la entropía de permutación y la complejidad de Lempel-Ziv mostraron los mejores desempeños en altos desbalances cuando se combinaron con la distancia de Levenshtein.

Los mejores resultados de la distancia de Levenshtein se deben a que esta característica es una medida para un conjunto de referencia de miARN, lo que permite medir con mayor precisión la pertenencia de cualquier secuencia a la clase positiva. Esta característica ha proporcionado una precisión muy alta en los clasificadores evaluados, lo cual es una de las contribuciones más importantes de esta tesis, debido a que la mayoría de los algoritmos disponibles tienen una tasa muy alta de falsos positivos. Además, ha mostrado robustez al desbalance, mejorando las predicciones incluso en los escenarios más adversos. Así, se demostró que la inclusión de una única característica del miARN maduro basada en la distancia de Levenshtein, puede mejorar hasta tres veces el desempeño en términos de F_1, manteniendo la precisión en un 90 % aún para desbalances de 1:10000.

Además, se desarrolló miRe2e, el primer algoritmo end-to-end de predicción de pre-miARNs en genomas crudos utilizando técnicas de aprendizaje profundo y procesamiento del lenguaje natural. Este modelo tiene dos grandes ventajas sobre los otros métodos existentes. Por un lado, es capaz de recibir datos de todo el genoma sin ningún preprocesamiento o predicción de estructura secundaria. Así, es posible minimizar el impacto de las decisiones humanas en la etapa de extracción de características, mejorando la reproducibilidad y replicabilidad de los resultados. Por otro lado, miRe2e puede identificar todas las secuencias de pre-miARN dentro de un genoma con una muy alta precisión y sensibilidad. Además, se ha demostrado que es muy poco afectado por el alto desbalance de clases que existe dentro de un genoma completo, entre los posibles nuevos pre-miARN y la enorme cantidad de secuencias negativas. Así, en los experimentos realizados con el genoma humano se pudo descubrir de manera efectiva nuevos pre-miARN.

Sección 5

Publicaciones

A continuación se listan todos los trabajos relacionados con la predicción de microARN en los que se participó durante el desarrollo del doctorado.

Publicaciones en revistas

- Stegmayer G., Di Persia L.E., Rubiolo M., Gerard M., Pividori M., Yones C., Bugnon L.A., Rodriguez T., **Raad J.**, Milone D.H. Predicting novel microRNAs: a comprehensive comparison of machine learning approaches. *Briefings in bioinformatics* 20(1), 1607-1620 (IF 9,101) (2018).

- Bugnon L.A., Yones C., **Raad J.**, Milone D.H., Stegmayer G. Genome-wide hairpins datasets of animals and plants for novel miRNA prediction. *Data in brief* 25(8), 104209 (IF 1,13) (2019)

- **Raad, J.**, Stegmayer, G., & Milone, D. H. Complexity measures of the mature miRNA for improving pre-miRNAs prediction. *Bioinformatics* 36(8), 2319–2327 (IF 6,937) (2020).

- Bugnon L.A., Yones C., **Raad J.**, Gerard M., Rubiolo M., Merino G., Pividori M., Di Persia L.E., Milone D.H., Stegmayer G. DL4papers: a deep learning approach for the automatic interpretation of scientific articles. *Bioinformatics* 36(11), 3499–3506 (IF 6,937) (2020)

- Merino G., **Raad J.**, Bugnon L.A., Yones C., Kamenetzky L., Claus J., Ariel F, Milone D.H., Stegmayer G. Novel SARS-Cov-2 encoded small RNAs in the passage to humans. *Bioinformatics* 36(24), 5571–5581 (IF 6,937) (2020)

- Yones C., **Raad J.**, Bugnon L.A., Milone D.H., Stegmayer G. High precision in MicroRNA prediction with Deep Learning: a novel approach based on Convolutional Deep Residual Networks for genome-wide data. *Computers in Biology and Medicine* 134, 104448 (IF 4,589) (2021)

- **Raad, J.**, Bugnon. L.A., Milone, D. H. & Stegmayer, G. miRe2e: a full end-to-end deep model based on Transformers for prediction of pre-miRNAs from raw genome-wide data. *Bioinformatics* (En revisión).

Trabajos en eventos científicos

- **Raad, J.**, Stegmayer, G., & Milone, D. H. Complexity measures of the mature miRNA for improving pre-miRNAs prediction. In Proc. of *10th Argentinian Conference on Bioinformatics and Computational Biology, A2B2C.*, Buenos Aires (2019)

Bibliografía

Amin, N. *et al.* (2019). Evaluation of deep learning in non-coding RNA classification. *Nature Machine Intelligence*, **1**(5), 246–256.

Bandt, C. and Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, **88**(17), 174102.

Bartel, D. P. (2009). MicroRNAs: target recognition and regulatory functions. *cell*, **136**(2), 215–233.

Bartel, D. P. (2018). Metazoan microRNAs. *Cell*, **173**(1), 20–51.

Batuwita, R. and Palade, V. (2009). micropred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, **25**(8), 989–995.

Bengio, Y. *et al.* (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(8), 1798–1828.

Billoud, B. *et al.* (2013). Computational prediction and experimental validation of microRNAs in the brown alga Ectocarpus siliculosus. *Nucleic Acids Research*, **42**(1), 417–429.

Brennecke, J. *et al.* (2005). Principles of microRNA–target recognition. *PLoS biology*, **3**(3), e85.

Bugnon, L. A. *et al.* (2020a). Genome-wide discovery of pre-miRNAs: comparison of recent approaches based on machine learning. *Briefings in Bioinformatics*.

Bugnon, L. A., Yones, C., Milone, D. H., and Stegmayer, G. (2020b). Genome-wide discovery of pre-mirnas: comparison of recent approaches based on machine learning. *Oxford Briefings in Bioinformatics*.

Chaabane, M. *et al.* (2019). circDeep: deep learning approach for circular RNA classification from other long non-coding RNA. *Bioinformatics*, **36**(1), 73–80.

Chen, L. *et al.* (2018). Trends in the development of miRNA bioinformatics tools. *Briefings in Bioinformatics*.

D. Gusev, V., A. Nemytikova, L., and A. Chuzhanova, N. (1999). On the complexity measures of genetic sequences. *Bioinformatics*, **15**(12), 994–999.

de ON Lopes, I. *et al.* (2014). The discriminant power of RNA features for pre-miRNA recognition. *BMC bioinformatics*, **15**(1), 124.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, **7**(Jan), 1–30.

Devlin, J. *et al.* (2018a). Bert: Pre-training of deep bidirectional transformers for language understanding. cite arxiv:1810.04805Comment: 13 pages.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018b). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, J. *et al.* (2010). MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC bioinformatics*, **11**(11), S11.

Dosovitskiy, A. *et al.* (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

Eraslan, G. *et al.* (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, **20**(7), 389–403.

Friedman, R. C. *et al.* (2009). Most mammalian mRNAs are conserved targets of microrna. *Genome research*, **19**(1), 92–105.

Grassberger, P. (1991). Information and complexity measures in dynamical systems. In *Information dynamics*, pages 15–33. Springer.

Gudyś, A. *et al.* (2013). HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC bioinformatics*, **14**(1), 83.

He, K. *et al.* (2016). Identity mappings in deep residual networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham. Springer International Publishing.

Hertel, J. and Stadler, P. F. (2006). Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, **22**(14), e197–e202.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, **18**(7), 1527–1554.

Hofacker, I. L. (2003). Vienna rna secondary structure server. *Nucleic acids research*, **31**(13), 3429–3431.

Jiang, P. *et al.* (2007). MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic acids research*, **35**(suppl_2), W339–W344.

Jurtz, V. I. *et al.* (2017). An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, **33**(22), 3685–3690.

Kleftogiannis, D., Korfiati, A., Theofilatos, K., Likothanassis, S., Tsakalidis, A., and Mavroudi, S. (2013). Where we stand, where we are moving: surveying computational techniques for identifying mirna genes and uncovering their regulatory role. *Journal of biomedical informatics*, **46**(3), 563–573.

Lassmann, T. and Sonnhammer, E. L. (2005). Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC bioinformatics*, **6**(1), 298.

LeCun, Y. *et al.* (2015). Deep learning. *Nature*, **521**(7553), 436–444.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Lewis, B. P. *et al.* (2003). Prediction of mammalian microRNA targets. *Cell*, **115**(7), 787–798.

Lewis, B. P. *et al.* (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *cell*, **120**(1), 15–20.

Li, L. *et al.* (2010). Computational approaches for microRNA studies: a review. *Mammalian Genome*, **21**(1-2), 1–12.

Lorenz, R. *et al.* (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, **6**(1), 26.

Mathelier, A. and Carbone, A. (2010). MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, **26**(18), 2226–2234.

Nan, F. and Adjeroh, D. (2004). On complexity measures for biological sequences. pages 522– 526.

Ng, K. L. S. and Mishra, S. K. (2007). De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, **23**(11), 1321–1330.

Park, S. *et al.* (2017). Deep recurrent neural network-based identification of precursor microrna s. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2895–2904.

Saito, T. *et al.* (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, **10**(3).

Senior, A. W. *et al.* (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, **577**(7792), 706–710.

Seo, S. *et al.* (2018). DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, **34**(13), i254–i262.

Shukla, V. *et al.* (2017). A compilation of web-based research tools for miRNA analysis. *Briefings in functional genomics*, **16**(5), 249–273.

Stegmayer, G. *et al.* (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*, **20**(5), 1607–1620.

Tang, X. and Sun, Y. (2019). Fast and accurate microRNA search using CNN. *BMC Bioinformatics*, **20**(S23).

Trieu, H.-L. *et al.* (2020). DeepEventMine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, **36**(19), 4910–4917.

Tsubaki, M. *et al.* (2018). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**(2), 309–318.

Vaswani, A. *et al.* (2017). Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wheeler, B. M. *et al.* (2009). The deep evolution of metazoan microRNAs. *Evolution & development*, **11**(1), 50–68.

Xue, C. *et al.* (2005). Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, **6**(1), 310.

Yones, C. *et al.* (2015). miRNAfe: a comprehensive tool for feature extraction in microRNA prediction. *Biosystems*, **138**, 1–5.

Yones, C. *et al.* (2020). HextractoR: an r package for automatic extraction of hairpins from genome-wide data. *bioRxiv*.

Yousef, M. *et al.* (2006). Combining multi-species genomic data for microRNA identification using a naive bayes classifier. *Bioinformatics*, **22**(11), 1325–1334.

Zeng, H. *et al.* (2016). Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, **32**(12), i121–i127.

Zheng, X. *et al.* (2019). Nucleotide-level convolutional neural networks for pre-miRNA classification. *Scientific Reports*, **9**(1).

Ziv, J. and Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, **24**(5), 530–536.

Zytnicki, M. *et al.* (2008). Darn! a weighted constraint solver for RNA motif localization. *Constraints*, **13**(1-2), 91–109.

Apéndices

Contribuciones

Complexity measures of the mature miRNA for improving pre-miRNAs prediction.

En este trabajo se presentaron las nuevas características basadas en medidas de complejidad del miARN maduro. Esta publicación corresponde con las propuestas originales para la primera etapa de la metodología desarrollada en la tesis. En este trabajo mi contribución fue el desarrollo de la idea original, la implementación de las características, la ejecución de los experimentos y la redacción del manuscrito.

miRe2e: a full end-to-end deep model based on Transformers for prediction of pre-miRNAs from raw genome-wide data

En este trabajo se presentó el primer algoritmo de aprendizaje profundo de extremo a extremo para la predicción de pre-miARNs en genoma completo. Esta publicación corresponde a un desarrollo original para la segunda etapa de la metodología desarrollada en la tesis. En este trabajo me encargué de la revisión del estado del arte, de la propuesta y desarrollo de la idea original, del diseño e implementación del algoritmo, de la validación del algoritmo de predicción, de la comparación con trabajos relacionados y de la escritura del manuscrito.

High precision in MicroRNA prediction with Deep Learning: a novel approach based on Convolutional Deep Residual Networks for genome-wide data

En este trabajo se presentó un algoritmo de predicción de pre-miARNs basado en redes convolucionales profundas. En esta publicación mi contribución fue la colaboración en el diseño de los experimentos a genoma completo y en el diseño de la base de datos. Además realicé el análisis de los resultados de los miARN maduros predichos por la red y colaboré con la redacción del documento.

Novel SARS-Cov-2 encoded small RNAs in the passage to humans

En este trabajo se realizó la predicción de 6 nuevos pre-miARNs en el virus Sars-Cov-2 y se mostró como la mutación en sus miARN maduros pueden haber sido claves para su capacidad de infectar humanos. En esta publicación aporté mi experiencia en métodos predicción de pre-miARN y características del miARN maduro. Además realicé los alineamientos entre secuencias candidatas y los genomas de otras especies para evaluar la conservación de las mismas, como así los alineamientos múltiples de sus miARN maduros y el análisis de sus posibles mutaciones. Finalmente redacté la sección métodos de alineamientos de secuencias y realicé la revisión del manuscrito.

DL4papers: a deep learning approach for the automatic interpretation of scientific articles

En este trabajo se presentó un algoritmo de aprendizaje profundo para la interpretación automática de artículos científicos. En esta publicación colaboré debido a mi experiencia en redes recurrentes y Transformers en el análisis y selección de los diferentes embedding para el texto. Además realicé los experimentos para comparar el desempeño del modelo para cada uno de los embedding propuestos. Finalmente realicé la revisión del manuscrito.

Genome-wide hairpins datasets of animals and plants for novel miRNA prediction

En este trabajo se realizó un análisis y preparación de bases de datos de horquillas de diferentes genomas completos de animales y plantas, diseñadas para la predicción de nuevos pre-miARNs. Mi contribución fue la participación en el diseño y discusiones sobre la estructura secundaria del miARN y en el asesoramiento respecto a las diferencias entre la biogénesis de pre-miARNs en animales y plantas. Además realicé la redacción de su correspondiente sección en el texto. Finalmente participé en la revisión del manuscrito.

Predicting novel microRNAs: a comprehensive comparison of machine learning approaches

En este trabajo se realizó una revisión del estado del arte respecto a los algoritmos de predicción de pre-miARNs de los últimos 10 años, incluyendo un diseño experimental y comparación de todos los métodos con los mismos datos y en las mismas condiciones. Mi contribución a este trabajo fue la realización de los experimentos para distintos desbalances y varias técnicas de sobremuestreo para el clasificador de Naive Bayes. Además realicé la escritura de la descripción del correspondiente clasificador y la revisión del manuscrito.

Complexity measures of the mature miRNA for improving pre-miRNAs prediction

Complexity measures of the mature miRNA for improving pre-miRNAs prediction

Jonathan Raad*[1], Georgina Stegmayer[1], and Diego H. Milone[1]

[1]Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Ciudad Universitaria UNL, (3000) Santa Fe, Argentina.

Abstract

Motivation: The discovery of microRNA (miRNA) in the last decade has certainly changed the understanding of gene regulation in the cell. Although a large number of algorithms with different features have been proposed, they still predict an impractical amount of false positives. Most of the proposed features are based on the structure of precursors of the miRNA (pre-miRNA) only, not considering the important and relevant information contained in the mature miRNA. Such new kind of features could certainly improve the performance of the predictors of new miRNAs.

Results: This paper presents three new features that are based on the sequence information contained in the mature miRNA. We will show how these new features, when used by a classical supervised machine learning approach as well as by more recent proposals based on deep learning, improve the prediction performance in a significant way. Moreover, several experimental conditions were defined and tested in order to evaluate the novel features impact in situations close to genome-wide analysis. The results show that the incorporation of new features based on the mature miRNA allow to improve the detection of new miRNAs independently of the classifier used. The source code is freely available for academic use under GPL license at `https://sourceforge.net/projects/sourcesinc/files/cplxmirna/`.

keywords: microRNA, Feature extraction

1 Introduction

In the recent decades, the discovery of new non-coding RNA molecules has changed the understanding of gene regulation in the cell. One of those molecules that caught most of the attention of the scientific community has been the microRNA (miRNA), due to its importance in the promotion or inhibition of several diseases (Bartel, 2004; Takahashi *et al.*, 2015). The miRNAs are small RNA molecules, approximately 21 bases long, which regulate gene expression in animal and plant cells through post-transcriptional control (Bartel, 2004). Given their proven role in promoting or inhibiting genes, the discovery of more miRNAs is of high interest today. Up to date, there are 38,589 miRNAs in miRBase v22[1]. Small RNA deep sequencing datasets have been used in order to support their validity. The read mapping patterns provided strong support for between 20% to 65% (depending on the species) microRNA annotations (Kozomara *et al.*, 2019). It is expected that the number of miRNAs continues growing. In fact, it has been increasing with every new release of miRBase: in v19 there were 25,141 and 30,582 in v21.

In a genome, the miRNAs are stored inside precursors that allow their recognition (Bartel, 2004). Precursors of miRNAs (pre-miRNAs) are molecules of 100 bases long approximately, which have a stem-loop structure. Experimental methods for detecting pre-miRNAs can be performed with different techniques, such as quantitative real-time PCR (qPCR), microarray and deep sequencing. These

*jraad@sinc.unl.edu.ar

[1]http://www.mirbase.org/

techniques present some practical difficulties when evaluating a large number of candidates. First, both qPCR and microarray suffer from low specificity and need extensive normalization (Baker, 2010; Dong *et al.*, 2013). In addition, prior knowledge is needed for the design of primers for qPCR and target sequences for microarrays, which does not allow finding novel pre-miRNAs (Pritchard *et al.*, 2012). In the case of deep sequencing, prior knowledge is not necessary but this technique is hampered by the need of extensive downstream computational analysis (Demirci *et al.*, 2017). Due to these technical and practical difficulties in detecting pre-miRNAs, computational methods have been playing an increasingly important role for their prediction (Li *et al.*, 2010; de ON Lopes *et al.*, 2014).

Among computational methods, two main prediction strategies can be considered: rule-based (RB) and machine learning (ML) based algorithms. RB algorithms evaluate measures of each sequence against reference values obtained from known pre-miRNAs. Two examples of RB tools are (Mathelier and Carbone, 2010; Friedländer *et al.*, 2011). ML based algorithms require a training step on features calculated from known pre-miRNAs and a negative set. Several RB and ML based tools were revised in (Bortolomeazzi *et al.*, 2017). The adjustment of parameters for each methods can be done automatically (by grid search or learnt from data) or manually. For example, if a given distance is calculated among sequences, a threshold must be set. If the prediction method is used with other data (for example, a newer version of miRBase), this threshold will have to be manually adjusted again. Instead, a threshold (or any other parameter) that can be automatically learnt according to data distribution, as in ML, could be used with these and with other newer data, without requiring a manual readjustment by an expert. A large number of approaches based on ML have emerged recently, for example with random forests (Vitsios *et al.*, 2017), support vector machines (Tseng *et al.*, 2017), graph based semi-supervised learning model (Yones *et al.*, 2018), and deep neural architectures (Bugnon *et al.*, 2019). Most of them propose novel ML models using a standard feature extraction. Differently, in this work we will propose novel features and will test them with standard ML classifiers. Many reviews have analysed the advantages of ML tools. For example (Chen *et al.*, 2018) reviews 20 miRNA bioinformatics tools published before 2018, where 11 out of 20 are ML-based. It concluded that classic ML methods, such as support vector machines, are still popularly used in the miRNA field, while novel and more advanced deep learning methods are beginning to appear. In (Stegmayer *et al.*, 2018), 29 pre-miRNA ML-based prediction tools published in the last 10 years are included. (Morgado and Johannes, 2017), affirmed that ML models can capture more general features that other approaches, which allows them to better detect miRNA sequences and precursors, even those with low similarity to the reference set. In (Liu, 2017) is analyzed in detail a web-server that can construct a very large variety of ML predictors for miRNAs. It is based on the fact that ML learning techniques are playing key roles in this field nowadays, but they can be cumbersome to build and use. Thus, this web server has been proposed to automatically complete the main steps for constructing a ML-predictor. A recent study (Demirci *et al.*, 2017) has shown that the computational prediction of pre-miRNAs is yet far-away from being satisfactory solved.

In order to find new candidates for pre-miRNA, structural and sequence characteristics of hairpins in a genome have to be extracted to train an ML classifier (Li *et al.*, 2010; de ON Lopes *et al.*, 2014; Shukla *et al.*, 2017). In the literature, many different features sets have been proposed, which mostly describe information of the structure of the pre-miRNA inspired by the action of Drosha (de ON Lopes *et al.*, 2014). However, although the microprocessor can takes a leading role in choosing which RNA precursors encode a miRNA, the specificity of the subsequent processes can impose additional restrictions on those hairpins that will eventually become mature miRNA (Bartel, 2018). In addition, in different studies it has been found that the selectivity of the miRNA for the target mRNA is defined by the sequence of the corresponding mature miRNA (Friedman *et al.*, 2009; Lewis *et al.*, 2005; Brennecke *et al.*, 2005; Bartel, 2009). Specifically, the mature miRNA contains two areas of union with the target sequence called seed and complementary site (Friedman *et al.*, 2009). Due to the importance that the seed has in the sequence function, the mature miRNAs can be classified on the basis of the presence of identical seed sequences into groups called miRNA families (Lewis *et al.*, 2003). In fact, some authors have proposed automatic classifiers for miRNAs families (Zou *et al.*, 2014). Therefore, given that important information is codified in the mature region, the secondary structure of the precursor by itself might not be sufficient to differentiate a true pre-miRNA from other hairpins. Our hypothesis is that the main difficulty in separating both classes is due to the omission of relevant information regard-

ing the mature miRNA sequence in the description (feature extraction process) of the pre-miRNAs. This fact is especially notable in the prediction of novel precursors, where the features are extracted mainly from the sequences structure. A typical example of this kind of standard features (SF) is the triplets representation (Xue *et al.*, 2005), which considers the structural composition of three adjacent nucleotides and the middle base to build a vector with 32 elements. Other examples are the number of internal loops and their length (Yousef *et al.*, 2006), the z-score of the minimum free energy (Hertel and Stadler, 2006), the dinucleotide proportion (Batuwita and Palade, 2009), base pair proportion, G+C content in the terminal loop (de ON Lopes *et al.*, 2014), Shannon's entropy (zQ), base pair propensity (zP) (Ng and Mishra, 2007) and base pair distance (zD) (Ding *et al.*, 2010). Although many features have been proposed, those are mostly based on the secondary structure of pre-miRNA or the relative frequencies of dinucleotides, trinucleotides and motifs in these sequences (de ON Lopes *et al.*, 2014; Yones *et al.*, 2015). These features have been performing quite well on current classifiers (Stegmayer *et al.*, 2018). However, it can be stated that these SF do not allow to represent nor to preserve the information regarding the order in which these triads and motifs are present in the sequence, losing valuable information regarding the coding of the mature miRNA within a sequence itself.

In this work, we propose three new features that take particularly into account the order in which the nucleotides are presented in the mature miRNA, which can effectively improve the sequence representation. We will show how these novel features can improve the prediction of novel pre-miRNAs, independently of the classifier. One of the proposed features is based on the Levenshtein distance. The rationale behind it is that candidate sequences to be new miRNAs should be very similar in the region encoding the mature, and Levenshtein distance can measure it in terms of nucleotides editions. This distance has been used in other areas of bioinformatics like sequence alignment, and also to estimate the proximity between sequences (Zytnicki *et al.*, 2008; Lassmann and Sonnhammer, 2005; Billoud *et al.*, 2013). The first algorithm for global alignment was proposed as a modification of the Levenshtein distance (Needleman and Wunsch, 1970), where the problem was formulated in terms of maximizing the similarity between sequences. Subsequently, different approaches appeared such as local and semi-global alignment. The local alignment seeks to align dissimilar sequences that contain small regions of similarity in large contexts (Polyanovsky *et al.*, 2011). The semi-global alignments are used to align short sequences with large sequences, through a global alignment of the first and a local alignment of the second one (Brudno *et al.*, 2003). However, the reason why the Levenshtein distance was chosen in our work is for obtaining a numerical measure to better quantify the distance (and not maximizing the similarity) between two short sequences (mature miRNAs). Therefore, due to the conservation and the evolution of miRNAs (Wheeler *et al.*, 2009), we will show how the chains that codify the mature miRNA of possible pre-miRNA sequences are closer in this space than those that do not encode miRNAs. This way it is possible to calculate, for each candidate sequence, a distance to labeled pre-miRNAs in order to evaluate how close each candidate is to these pre-miRNA samples. Differently from (Mathelier and Carbone, 2010), where the Levenshtein distance is used as a direct calculation of the edition errors with a threshold for eliminating sequences as a first step of the processing, in our work we build a statistic that can estimate the belonging of the candidate sequence to the set of positive class examples. This way, the Levenshtein distance as a feature is more general and applicable to any species, and can be used by any classifier. The second and third proposed features were inspired, from the point of view of the information theory, considering the randomness of a sequence that would encode a mature miRNA in the hairpin. In addition, it is known that certain mature regions have specific motifs that define their functionality and the belonging to a specific miRNA family (Bartel, 2018, 2009). In order to quantify this fact, we propose a permutation entropy (Bandt and Pompe, 2002) feature and a measure of the Lempel-Ziv complexity (Ziv and Lempel, 1978) of the sequences. We have measured the performance of these new features when used by classical supervised machine learning approaches such as Naive Bayes (NB), Random Forest (RF), k-nearest neighbor (KNN) and more recent proposals based on deep neural networks (DNN).

2 Novel features based on complexity measures

2.1 Levenshtein distance

During evolution, many miRNAs were mostly preserved among different species, sometimes suffering modifications that resulted in new miRNAs. Despite these modifications over time, the preservation of specific sequences such as the seeds of mature miRNAs has been studied, defining functionality as well as the belonging to a specific family (Bartel, 2018). This leads us to believe that the sequences that can be candidates to new pre-miRNAs should be very similar in the region encoding a mature. In other words, as a result of evolution, one would expect to have a small nucleotide edit distance in those sequences that can effectively encode miRNAs.

The Levenshtein distance, L, also known as edit distance between strings, is defined as the minimum number of operations (insertions, deletions or substitutions) required to transform one string into another one (Levenshtein, 1966). This distance between two strings $\boldsymbol{x}$ and $\boldsymbol{y}$, of lengths $|\boldsymbol{x}|$ and $|\boldsymbol{y}|$, can be calculated according to Algorithm 1. The algorithm begins verifying that both chains have a length greater than zero (line 1). If either of the two does not satisfy the condition, the algorithm returns the length of the other chain (line 2), that is, the number of insertions necessary to build it from an empty chain. If both chains satisfy the previous condition, a matrix D of $|\boldsymbol{x}| + 1$ rows and $|\boldsymbol{y}| + 1$ columns is created where the first row is initialized with values from 0 to $|\boldsymbol{x}|$, and the first column from 0 to $|\boldsymbol{y}|$ (lines 4 and 5). Then for each element $d_{i,j}$ in the matrix D, it is verified if x_i is equal to y_j. If this equality is satisfied, no editing operation is required. Otherwise, since one string chain can be obtained in different ways from the other one, we want to find the strings that require the fewest editing operations in relation to the other one(that is, the minimum edit distance between them). For this purpose, the minimum value of the three possible string operations is obtained in line 9, where the $d_{i-1,j} + 1$, $d_{i,j-1} + 1$ and $d_{i-1,j-1} + c$ corresponding to the operations of insertion, deletion and substitution, respectively. The variable c corresponds to a substitution cost. It is calculated in line 8, where $\delta(x_i, y_j)$ is the Dirac delta. The cost c is equal to 0 when both characters are equal, and 1 otherwise. It must be noted that for insertion and deletion, cost is always 1. Finally, the value found in last element of D, $d_{|\boldsymbol{x}|,|\boldsymbol{y}|}$, is assigned as the Levenshtein distance between the analyzed chains (line 10). Since this measure adds insertion steps when two chains have different lengths, it is necessary to define a way to be able to compare the distances between pairs of candidates, regardless their individual lengths are different. That is why in line 10 each distance is adjusted by subtracting the absolute difference of the lengths of the strings under analysis.

In order to be able to calculate L as a feature for each hairpin sequence, and since L is a distance between two elements, it is necessary to have a reference set for comparison. Let be $\mathcal{A}$ the set with the miRNA matures $\boldsymbol{a}_k$. Let $\boldsymbol{a}_\ell$ an element of $\mathcal{A}$ for which we wants to obtain the L feature. Then, the median of the distance of $\boldsymbol{a}_\ell$ to all the other elements of the set can be as feature of $\boldsymbol{a}_\ell$, that is

$$L_{\mathcal{A} \setminus \boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \text{med}_{\forall k \neq \ell}\{\boldsymbol{a}_k, \boldsymbol{a}_\ell\}, \tag{1}$$

where $\mathcal{A} \setminus \boldsymbol{a}_\ell$ is the set $\mathcal{A}$ without the element $\boldsymbol{a}_\ell$. Then, each candidate can have its mature coding in different regions (5p or 3p), it is necessary to extract two chains $\boldsymbol{a}_\ell^{5p}$ and $\boldsymbol{a}_\ell^{3p}$. Thus, two L measures for each $\boldsymbol{a}_\ell$ are obtained and the maximum edit value between both $L_{\mathcal{A} \setminus \boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p})$ and $L_{\mathcal{A} \setminus \boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})$ is selected as the final $L(\boldsymbol{a}_\ell)$. That is, the L feature is not based on the distance to the primary mature strand alone, but also to its corresponding complementary star strand as well. When the distance with respect to both strands is calculated, selecting afterwards the maximum, both strands must comply with a certain minimum distance to the known miRNAs so that the L feature evidences a miRNA. That is to say, this way, none of the two strands has an excessive distance to the known pre-miRNAs.

2.2 Permutation entropy

The section in the hairpin that encodes the mature miRNA contains specific patterns of the nucleotides order in its seed and in its complementary region (Friedman *et al.*, 2009; Lewis *et al.*, 2005; Bartel, 2009). Thus, it can be expected that pre-miRNAs have less randomness in that section than any other

Algorithm 1: Levenshtein distance

Input : $\boldsymbol{x}$, $\boldsymbol{y}$ RNA sequence strings
Output: L Levenshtein distance

if $|\boldsymbol{x}||\boldsymbol{y}| = 0$ **then**
  $L \leftarrow \max\{|\boldsymbol{x}|, |\boldsymbol{y}|\}$
else
  $d_{i,0} \leftarrow\ i\ \ \forall i$
  $d_{0,j} \leftarrow j\ \ \forall j$
  for $i \leftarrow 1$ **to** $|\boldsymbol{x}|$ **do**
    for $j \leftarrow 1$ **to** $|\boldsymbol{y}|$ **do**
      $c \leftarrow 1 - \delta(x_i, y_j)$
      $d_{i,j} \leftarrow \min\ \{d_{i-1,j} + 1,\ d_{i,j-1} + 1,\ d_{i-1,j-1} + c\}$
  $L \longleftarrow d_{|\boldsymbol{x}|,|\boldsymbol{y}|} - ||\boldsymbol{x}| - |\boldsymbol{y}||$
return L

sequences. Therefore, a measure capable of quantifying such randomness in sequence patterns could be useful to detect the true pre-miRNAs.

The Shannon entropy is widely used to measure the randomness of a sequence: the more random, the larger the entropy (Shannon, 2001). The drawback of this approach when analyzing miRNA sequences is that the information of the internal order of the nucleotides is lost when calculating the relative frequencies. To solve this, Bandt and Pompe in (Bandt and Pompe, 2002) proposed a new coding based on permutation patterns in the sequence, where the entropy is estimated from the relative frequencies of these patterns. The measure was called permutation entropy (PE). In this case, the probability distribution of $\boldsymbol{x}$ was replaced by the relative frequencies $p_{\boldsymbol{\pi}}$ of all possible patterns $\boldsymbol{\pi}$ that can be found within $\boldsymbol{x}$.

When working with PE, it is necessary to previously choose the length of the patterns to be permuted. This parameter is called order N. Thus, defined the order, $N!$ patterns $\boldsymbol{\pi}$ of length N are obtained. For example, selecting $N = 3$, then 6 possible patterns are possible: (1,2,3) (1,3,2) (2,1,3) (3,2,1) (3,1,2) (2,3,1). If the frequencies of these patterns are calculated in $\boldsymbol{x}$, then the corresponding PE can be estimated as

$$PE_N(\mathbf{x}) = -\sum_{i=1}^{N!} p_{\boldsymbol{\pi}_i} \cdot \log_2(p_{\boldsymbol{\pi}_i}), \tag{2}$$

When N is too small, relevant information from the system dynamics cannot be captured. On the other hand, if N is very large, the sequence will require a longer length in order to obtain a good estimation of the probability of each pattern. Therefore, as a practical rule (Bandt and Pompe, 2002), N must be selected in such a way that $N! \ll |\boldsymbol{x}|$. In the case of RNA sequences, they are encoded in an alphabet of 4 nucleotides that can form different combinations. In order to analyze as many combinations as possible, and due to the fact that the mature sequences have an approximate length of 25 nt, N should be just 2 or 3.

2.3 Lempel-ziv complexity

When observing the specificity of the mature sequence with respect to its corresponding target mRNA, from an information theory point of view, there must be syntactic rules that avoid any random mutation to modify their function. In other words, the coding of a mature sequence should be contained in a 'dictionary', so that more complex combinations of nucleotides are constructed from simpler combinations. Since the sequence of a mature must be encoded only by specific 'words', it is expected for those candidates that encode miRNA to have a smaller dictionary than those candidates that do

not. Therefore, it could be very useful to have a measure to quantify this complexity in a sequence of nucleotides.

The Lempel-Ziv (LZ) algorithm allows the calculation of such complexity in a finite sequence based on the analysis of its "production process" (Lempel and Ziv, 1976). Let $\boldsymbol{a}$ be a RNA sequence, which is composed of the 4 nucleotides. We define $\boldsymbol{a}(i,j)$ as a subsequence of $\boldsymbol{a}$ that is composed of the elements that are between the indices i and j. We say that $\boldsymbol{a}$ is reproducible from $\boldsymbol{a}(1, j)$, if $\boldsymbol{a}(j{+}1, |\boldsymbol{a}|)$ is a sub-word of $\boldsymbol{a}$ that is contained in $\boldsymbol{a}(1, j)$. Then, we say that $\boldsymbol{a}$ is producible from $\boldsymbol{a}(1, j)$, if we add a new element at the end of the sequence $\boldsymbol{a}$ that cannot be obtained by reproducing $\boldsymbol{a}(1, j)$. In other words, a chain $\boldsymbol{a}$ can be obtained from the extension of smaller chains by two processes: reproduction (when the extension is done by copying a substring of the smallest chain) or production (when the extension is done by a new substring that is not contained in the initial chain). For example, given the sequence ACACCA, we can obtain the dictionary A | C | AC | CA. Then, the sequence ACACCACAA is obtained by production when adding a new substring CAA that is not contained in the dictionary. However, the chain ACACCAAC is obtained from the original sequence ACACCA by reproduction of AC element.

If we concatenate all the processes by which the chain $\boldsymbol{a}$ can be formed, the history of its construction $H(\boldsymbol{a})$, is obtained. With this history, we can measure the complexity of such construction as the number of steps necessary to generate it. In addition, since it is possible to obtain a chain from another one in different ways, we are interested in finding the history that has the minimum necessary number of steps. If we consider each step of the process as reproduction or production, then $\boldsymbol{a}$ can be analyzed as a process of z steps $H(\boldsymbol{a}) = H_1(\boldsymbol{a})H_2(\boldsymbol{a})...H_z(\boldsymbol{a})$ with $h_0 \equiv 0$.

Then, let $|H(\boldsymbol{a})|$ be the number of steps in $H(\boldsymbol{a})$. The Lempel-Ziv complexity of a sequence $\boldsymbol{a}$ is thus defined as $\mathrm{lz}(\boldsymbol{a}) = \min\{|H(\boldsymbol{a})|\}$, regarding all the histories of $\boldsymbol{a}$. Then, to obtain a measure that is independent of the length of $\boldsymbol{a}$,

$$LZ(\boldsymbol{a}) = \frac{\mathrm{lz}(\boldsymbol{a}) \log_4 |\boldsymbol{a}|}{|\boldsymbol{a}|}, \tag{3}$$

where 4 in the base of the logarithm represents the number of nucleotides.

3 Materials, measures and experimental setup

3.1 Datasets

For this study we have created a number of datasets of varying ratios of class imbalance, testing pre-miRNA predictors with and without the proposed new features. We have used an already available public dataset (Gudyś *et al.*, 2013), which provides negative and positive samples of all known pre-miRNAs in miRBase (Kozomara and Griffiths-Jones, 2010) for *Homo sapiens* (1,406 positives and 81,228 negatives). The standard features are those used in the mostly cited works (see details in the Supplementary Material) (Stegmayer *et al.*, 2018; Jiang *et al.*, 2007; Gudyś *et al.*, 2013; Batuwita and Palade, 2009). The varying ratios of class imbalance allows to evaluate the robustness of the new features in situations closer to those found in a real genome, where the number of positive miRNAs is very low with respect to the number of hairpins without miRNA in the rest of a complete genome. For this purpose, datasets were generated by random sampling from 1:500 (1 positive in 500 negatives) to a very high imbalance 1:10,000 (1 positive in 10,000 negatives).

3.2 Performance measures

For performance evaluation, the following standard measures have been used

$$\text{Recall } s^+ = \frac{TP}{TP+FN}, \qquad \text{Precision } p = \frac{TP}{TP+FP},$$

$$\text{Specificity } s^- = \frac{TN}{TN+FP}, \qquad \text{F-measure } F_1 = 2\,\frac{s^+\,p}{p+s^-},$$

Matthew correlation coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

Kappa coefficient

$$\kappa = \frac{a - a_c}{1 - a_c},$$

where TP, TN, FP and FN are true positives, true negatives, false positives and false negatives, respectively; N is the total number of observations; $a = (TP + TN)/N$ is the standard accuracy and a_c is the accuracy by chance, that is, the one provided by a classifier assigning randomly a positive or negative label to each sample.

The true positives rate is measured with s^+, while the true negatives rate is measured with s^-. The precision p is key to evaluate the performance of a classifier in the context of large imbalances due to the impact of false positives. Although only a small fraction of the negatives are misclassified, it becomes a large number in comparison to the number of positives. This detail is fundamental when a realistic scenario is considered, where biologists need only a small set of candidates. Thus, F_1 becomes the best measure to compare classification methods in large class imbalances, combining s^+ and p through the harmonic mean. Furthermore, we used two more combined measures, MCC and κ, which are also used for imbalanced datasets.

3.3 Experimental setup

To calculate the features, the secondary structure of all sequences (positives and negatives) was predicted with RNAfold (Lorenz *et al.*, 2011), with 37°C and the remaining parameters by default. After that, the 5p and 3p chains were extracted with 40 nt length from the terminal loop. In this way, the specific position of the mature miRNA within the chain is not required. Thus, it is possible to calculate the feature without any additional information for unknown hairpins. This is important because different iso-miRs of the same chain can be generated depending on the position of the cut (Bartel, 2018).

The performance in each experiment is reported as the average value of 8 folds for the imbalances from 1:500 to 1:1,000, and 4 folds for the imbalances from 1:1,500 to 1:10,000, using the test partition only. This difference in the number of folds selected for each case is due to the decrease in the number of positives when the imbalance increases. To assess whether there is a statistically significant difference in the performance of the proposed sets of features, the Friedman test was performed for the F_1 measure with a significance level of $\alpha = 0.01$. Finally, to evaluate which features have statistically different performances, the Nemenyi post-hoc test was used (Demšar, 2006).

The LD feature must be calculated taking into account that the reference set (the positive pre-miRNAs) changes with each training partition. Therefore, only the mature miRNAs found in each training set $\mathcal{A}$ of each corresponding fold are used, thus avoiding introducing *a-priori* information from the corresponding test set. For the training sequences, the distance of each training sample $\boldsymbol{a}_\ell \in \mathcal{A}$ is calculated as $L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell) = \max\{L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{5p}), L_{\mathcal{A}\setminus\boldsymbol{a}_\ell}(\boldsymbol{a}_\ell^{3p})\}$. In the case of the test samples $\boldsymbol{t}_\ell$, all the sequences in the train set can be used and the feature is calculated as $L_{\mathcal{A}}(\boldsymbol{t}_\ell) = \max\{L_{\mathcal{A}}(\boldsymbol{t}_\ell^{5p}), L_{\mathcal{A}}(\boldsymbol{t}_\ell^{3p})\}$.

For the PE calculation, we selected $N = 2$ because this value showed the best performance in preliminary tests. We codified each nucleotide A, C, G, U with an integer from 1 to 4 according to its relative frequencies in the sequences. To combine the information from both chains 3p and 5p, we calculated PE for each one and selected the smallest one. That is, the PE of order 2 of each test candidate $\boldsymbol{t}$ is calculated as $PE_2(\boldsymbol{t}) = \min\{PE_2(\boldsymbol{t}^{5p}), PE_2(\boldsymbol{t}^{3p})\}$. In the same way the LZ of each test candidate $\boldsymbol{t}$ was calculated as $LZ(\boldsymbol{t}) = \min\{LZ(\boldsymbol{t}^{5p}), LZ(\boldsymbol{t}^{3p})\}$.

These new features were tested with Naive Bayes (NB), Random Forest (RF), k-nearest neighbor (KNN) and Deep Neural Network (DNN) classifiers. These classifiers have been chosen because they have provided the best performances in a very recent review study on pre-miRNA prediction approaches (Stegmayer *et al.*, 2018).

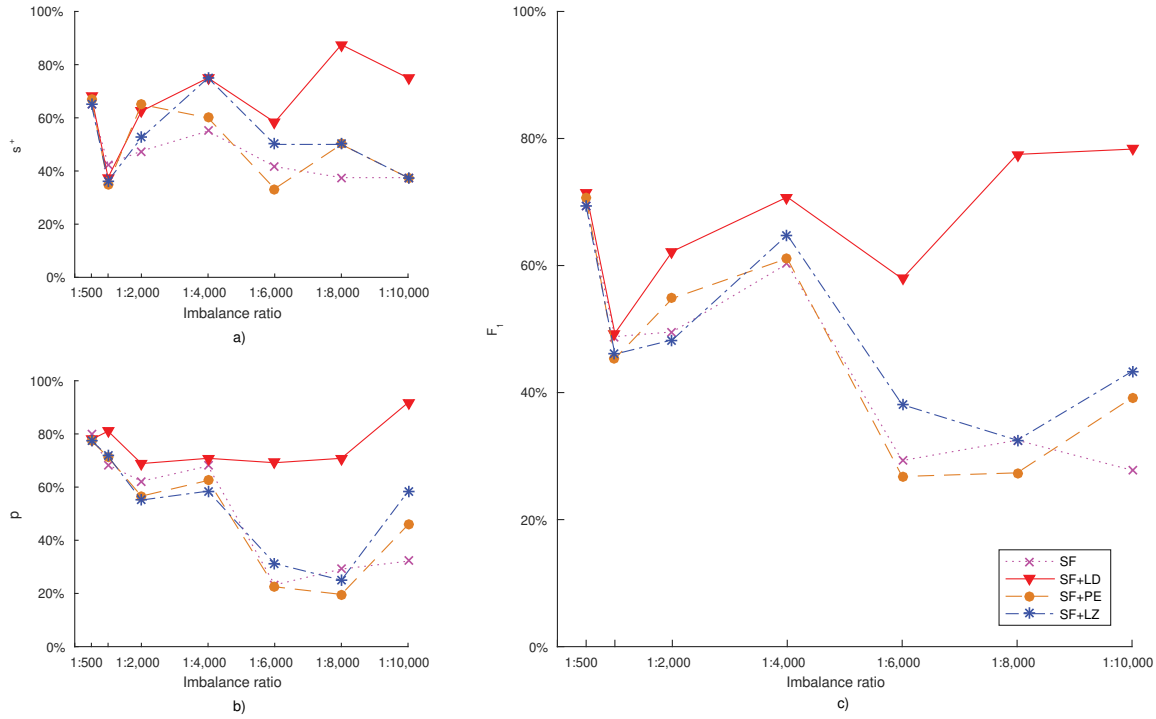

Figure 1: Results of deep neural networks (DNN) with standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). a) Sensibility, s^+; b) Precision, p; c) F_1 score.

NB classifiers are a family of probabilistic classifiers based on applying Bayes theorem (Webb, 2002) with strong assumptions of independence between the features. It calculates the probability that a given example belongs to a certain class, under the assumption that the features are conditionally independent given the class. A NB classifier can be seen as a probability function that assigns, to an unknown input $\boldsymbol{z}$, a class label $y(\boldsymbol{z})$, which is proportional to the product of the prior $p(y_j)$ and the conditional probability $p(\boldsymbol{z}_j|y_j)$. Gaussian distributions were used to train this model in our experiments. RF is an ensemble of decision trees (Breiman, 2001). A decision tree classifier is composed by a number of nodes starting from a root node. At each node, the training set is split into two non overlapping sets: for a selected feature, a threshold is chosen such that the sample is assigned to some set (Breiman, 2001). The tree is grown until a maximum depth. For the prediction of a new case, it is pushed down the tree and assigned the label of a terminal node. To avoid overfitting, bootstrap-aggregated (bagged) is used by combining the results of many trees. The final decision for an unknown input vector is made by taking the majority vote of the trees in the ensemble. We used 100 trees for all cases.

KNN is a method that stores all the training examples as the classification model, without building a parametric model. All computation occurs at testing time (without training). It does not fit a model to the data. KNN just looks for the k nearest neighbors in all the training dataset at testing time, and classifies according to the majority class of the neighbors (Webb, 2002). Therefore, the only parameter that needs to be set is the number of neighbors k. Euclidean distance was used with $k = 1$ for imbalances ratio less than 1:1,500 and $k = 3$ for the other ones.

A DNN can be built from several feedforward layers of nonlinear neurons. Layers that are commonly used in deep learning include latent variables organized layer-wise in deep generative models such as the restricted Boltzmann machines (RBM) (Fischer and Igel, 2012). After the unsupervised stage to train each RBM layer, a supervised training is applied to the full network. Therefore, this model uses

a hybrid learning approach. In this work, we used a network with 3 hidden layers and an output layer of 2 neurons. For imbalance of 1:500: 256, 128 and 16 neurons were used in each layer. For the second imbalance, 1:1,000: 256, 128, and 128 neurons were used in each layer. For the other cases: 256, 256, and 64 neurons were used for each layer. In all cases, the network was trained with cross entropy function and a batch size of 16. The optimization of these hyperparameter was done following (Stegmayer *et al.*, 2018).

4 Results and discussion

4.1 Classifiers and measures

Tables 1 to 4 present the results for each proposed new feature and the standard features (SF), for NB, RF, KNN and DNN classifiers, respectively. In each row, the performance of each classifier on a given imbalance, for all features, is reported according to MCC, κ and F_1. The best performance for each imbalance ratio and each measure is shown in bold.

Table 1 shows that, for NB with LD versus SF, the performance measures reflect consistently improvements for all imbalances. In particular, when LD are used, this classifier obtained the best rates in all imbalance cases. For the case where PE is used, improvements with respect to SF are found for all measures except for the imbalances of 1:2,000 and 1:4,000, where the performance remains the same. In the case of LZ, the same behavior is observed as in PE. In Table 2, when analyzing RF performance with the new features, all three performance measures show consistent results, that is, they improve the classifier performance in relation to SF alone. From 1:8,000 and on, all measures show that this classifier is highly affected by the imbalance. From the analysis of this table in a general way, it can be observed that the best results for each imbalance are distributed among the three features, but always exceeding SF in all cases and measures.

Table 3 shows KNN with LD versus SF. It can be seen here, again, that there is an improvement in performance when incorporating LD for imbalances less than 1:8,000. The only exception is for the imbalance of 1:4,000, where only F_1 shows an improvement in the classifier performance, while the other measures show the same result than SF alone. The other two features improve SF but only slightly and in some cases. At the highest imbalance point, KNN has an extremely poor performance, which is reflected by all measures. In Table 4, when analyzing the performance of DNN with LD versus SF, a significant improvement is observed in all the three measures and for all imbalances when the new LD feature is added to SF. For the case PE versus SF, it is observed that MCC and κ show improvements for the imbalances larger than 1:6,000. With F_1 the same improvement is found for all cases.

Finally, after a comprehensive analysis of all four tables in this section, it can be stated that, overall, improvements can be observed by all performance measures, consistently, and independently of the classifier used. It can be seen that RF and KNN show values equal to zero (or MCC of -1.0) for the largest imbalances. This is due to the bias generated by the *a-priori* probabilities of the classes, which causes the classifier to label the positive cases as part of the majority class (negative class). It is also observed that DNN achieved the highest performances for all imbalances and all features proposed, furthermore showing that these improvements are equally reflected by the three performance measures reported . For this reason, in the rest of this study, only this classifier will be used for the detailed analysis of the behavior of the proposed features. In addition, due to the fact that the three measures report a similar behavior, F_1 will be used from now on.

4.2 Detailed performance of novel features

Figure 1 shows a detailed analysis of the classification results for each of the new proposed features and SF, with DNN as classifier. The horizontal axis shows the imbalance ratio, while the vertical axis shows s^+, p and F_1, in Figures 1a, 1b and 1c, respectively. For more detailed information regarding the scores see Tables S1 to S4 (Supplementary Material). Since s^- has shown to be very close to 100% in all imbalances and for all features, it has not been included in the figure. This has happened because due to the high class imbalance, the negative class is the majority one and the easiest to

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.314	0.197	0.200	**0.324**	**0.207**	**0.210**	0.315	0.198	0.201	0.317	0.199	0.202
1:1,000	0.223	0.107	0.111	**0.234**	**0.115**	**0.119**	0.227	0.109	0.113	0.224	0.108	0.111
1:2,000	0.180	0.066	0.067	**0.184**	**0.069**	**0.071**	0.179	0.065	0.067	0.179	0.065	0.067
1:4,000	0.166	0.056	0.058	**0.180**	**0.066**	**0.067**	0.166	0.056	0.058	0.167	0.057	0.058
1:6,000	0.142	0.040	0.044	**0.164**	**0.052**	**0.057**	0.146	0.042	0.046	0.143	0.040	0.044
1:8,000	0.143	0.040	0.041	**0.178**	**0.061**	**0.063**	0.145	0.041	0.043	0.146	0.042	0.044
1:10,000	0.130	0.038	0.041	**0.153**	**0.052**	**0.061**	0.134	0.040	0.043	0.134	0.040	0.042

Table 1: Naive Bayes classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.650	0.630	0.633	0.664	0.646	0.646	0.664	0.646	0.646	**0.682**	**0.666**	**0.654**
1:1,000	0.602	0.532	0.510	**0.612**	**0.545**	**0.526**	0.498	0.456	0.453	0.591	0.518	0.492
1:2,000	0.418	0.298	0.279	**0.500**	**0.400**	0.372	0.447	0.333	0.311	**0.500**	**0.400**	**0.380**
1:4,000	0.447	0.333	0.266	0.387	0.261	0.208	**0.500**	**0.400**	**0.339**	0.387	0.261	0.194
1:6,000	-1.000	0.000	0.000	**0.289**	**0.154**	**0.125**	-1.000	0.000	0.000	-1.000	0.000	0.000
1:8,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000
1:10,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000

Table 2: Random Forest classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

detect, independently of the features employed.

Figure 1 clearly shows how the DNN classifier is capable of maintaining performance at increasing imbalances, and even increasing both s^+ (Figure 1a) and p (Figure 1b) when the new LD feature is used. This is a remarkable result, which has a direct impact in the impressive good performance of DNN with LD in F_1. In Figure 1c, when analyzing the performance of DNN with SF versus LD, it is observed that F_1 is significantly higher for all the imbalances when the new LD feature is used. For example, it can be seen that for the imbalances between 1:500 and 1:10,000, F_1 with SF goes down from almost 70% to around 20%. In this same imbalance range, however, DNN with LD goes up to almost 80%. It can also be noticed that the precision of the classifier increases very much with the incorporation of LD up to a very high level (higher than 90%) at the highest imbalance here studied. This is a very important result in practical terms, especially for imbalances closer to real cases where genome-wide data is used, because it assures to reduce remarkably the amount of false positives. Due to the fact that, in general terms, s^+ is also improved when LD is used, the F_1 increases in all cases as the imbalance increases. This is very interesting, since the ability to avoid false positives seems to be robust to the imbalance and the size of the positive set, without thereby influencing the detection of positives cases. When analyzing all the figures in a global way, an improvement of LD with respect to SF is observed for all the measures, which presents a clear trend to increase as the imbalance increases. The other features have more variable performance. In summary, it can be affirmed that a very important improvement in performance is obtained when using LD in the feature set, even at the highest imbalance.

An interesting point to discuss here is why LD shows such a robust behavior to imbalance. Generally, the algorithms for pre-miRNA prediction use public databases for training, which generates a bias towards previously known pre-miRNAs. Given that most of them have a stem-loop structure, and most of the features are based on that structure, with these standard features it is difficult to recognize

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.531	0.530	0.527	**0.568**	**0.568**	**0.574**	0.531	0.530	0.531	0.531	0.530	0.531
1:1,000	0.421	0.421	0.411	**0.441**	**0.441**	**0.447**	0.421	0.421	0.414	0.409	0.409	0.419
1:2,000	0.399	0.373	0.383	**0.494**	**0.476**	**0.478**	0.372	0.345	0.356	0.448	0.426	0.414
1:4,000	**0.592**	**0.518**	0.451	**0.592**	**0.518**	**0.476**	0.404	0.400	0.442	0.592	**0.518**	0.451
1:6,000	0.408	0.286	0.250	**0.577**	**0.500**	**0.367**	0.408	0.286	0.225	0.408	0.286	0.225
1:8,000	0.354	0.222	0.167	0.354	0.222	0.167	0.354	0.222	0.167	0.354	0.222	0.167
1:10,000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000	-1.000	0.000	0.000

Table 3: K-nearest neigbor classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

Imbalance	**SF**			**SF+LD**			**SF+PE**			**SF+LZ**		
ratio	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1	MCC	κ	F_1
1:500	0.704	0.702	0.695	**0.725**	**0.724**	**0.714**	0.697	0.697	0.707	0.704	0.702	0.693
1:1,000	0.499	0.492	0.488	**0.544**	**0.508**	**0.493**	0.472	0.451	0.453	0.483	0.464	0.461
1:2,000	0.508	0.506	0.496	**0.617**	**0.617**	**0.622**	0.506	0.490	0.548	0.495	0.494	0.483
1:4,000	0.564	0.564	0.603	**0.699**	**0.698**	**0.708**	0.600	0.600	0.611	0.699	**0.698**	0.648
1:6,000	0.400	0.400	0.293	**0.764**	**0.737**	**0.579**	0.333	0.333	0.268	0.463	0.461	0.381
1:8,000	0.320	0.316	0.325	**0.935**	**0.933**	**0.775**	0.408	0.400	0.274	0.408	0.400	0.325
1:10,000	0.320	0.316	0.278	**0.866**	**0.857**	**0.783**	0.612	0.545	0.392	0.612	0.545	0.433

Table 4: Deep neural networks classification results for standard features (SF), Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ). Results reported with Matthew correlation coefficient (MCC), Kappa coefficient (κ) and F_1 score.

possible new miRNAs that differ from the canonical ones. However, the inclusion of a sequence feature such as LD, calculated from the mature miRNA, is disruptive in this sense because it allows to take into account different information from the candidates, not related nor biased towards the structure alone. Thus, in a different space, generated by the novel features, the distances are different and the sequences that were not close according to standard features can be near now in the new space generated with the information of the mature miRNA. A second argument is that LD is not calculated only with the information of each candidate, but it is a distance of each sequence with respect to the whole reference set. A third point of view is that it can be said that this feature could be capable of obtaining a large robustness in front of candidates sequences that may have a more recent structure. This would be due to the incorporation of mature information that is complementary to the structure of each candidate. Thus, it could be possible to find new pre-miRNAs that differ from the canonical pre-miRNAs. One last interesting point to discuss is whether LD results can be biased towards larger miRNAs classes or families. Since in Eq. (1) LD is calculated as a statistic of the distances to each mature miRNAs of the training set, the choice of this statistic was not trivial. Firstly, the minimum has been chosen in order to avoid a possible bias towards the most numerous families. However, the results obtained showed a wide overlap of both classes, because the minimum considers only the most similar sequence. In contrast, the median is a more informative statistic because it uses the complete training set of known miRNAs. Thus, class distributions were shown to be more separated (see Figure S1 in the Supplementary Material).

For DNN with PE it is observed that F_1 is being improved in approximately a 10%, only at the largest imbalance here analyzed, where F_1 is almost 30% with SF, and almost 40% when PE is also used. The most important and remarkable improvement is observed in p at 1:10,000, where from around 30% it goes up to more than 45%. This suggests that this feature can effectively reduce the false positives, achieving an improvement of precision in very large imbalanced problems. In summary,

IR	SF	LD	PE	LZ	LD+PE	LD+LZ	PE+LZ	ALL
1:500	69.50	**71.44**	70.65	69.34	71.39	**71.68**	68.96	71.50
1:1,000	48.81	**49.33**	45.33	46.05	49.26	48.71	52.85	**53.85**
1:2,000	49.55	**62.22**	54.82	48.29	63.21	57.72	53.33	**65.34**
1:4,000	60.28	**70.78**	61.11	64.81	**78.28**	73.33	64.95	71.89
1:6,000	29.29	**57.92**	26.79	38.10	**61.67**	57.92	29.17	56.79
1:8,000	32.50	**77.50**	27.36	32.50	77.50	**85.00**	36.67	77.50
1:10,000	27.78	**78.33**	39.17	43.33	62.50	**70.00**	40.48	54.17

Table 5: F_1 results for different combinations of Levenshtein distance (LD), permutation entropy (PE) and Lempel-Ziv (LZ) with deep neural networks. Best results in bold for each table panel, individual (left) and combined (right) features.

it can be stated that PE can only improve the performance of DNNs just for highly imbalanced cases.

In the case of LZ, when analyzing the performance of DNN with SF, versus DNN with the incorporation of LZ, it is observed that F_1 is superior for the largest imbalance. It can also be seen that the improvement of F1 is due to by a slightly improvement of p and s^+. That is, LZ can probably serve to avoid false positives, especially when a negative class is extremely large with respect to the positive class. It can be stated, in summary, that LZ can have the capacity to improve the performance of a DNN for high imbalances, mainly thanks to the improvement of p.

4.3 Global performance of novel features

Table 5 shows the results with different combinations of the proposed features for DNN. In each row F_1 can be observed for the different sets of features, for each imbalance. It can be seen that LD improves the performance of the classifier in all cases, even for very high imbalances (1:10,000). Instead, LZ and PE individually do not improve the DNN performance. F_1 in those cases remains the same or quite similar to the SF case. Observing the different combinations of features for DNN, it can be noticed that F_1 improves for all cases in LD+PE with respect to SF. In addition, for the case of 1:2,000, 1:4,000 and 1:6,000, LD+PE combined achieve a larger performance than when used separately. For LD+LZ, F_1 improves in all cases with respect to SF (except for 1:1,000, where it remains almost the same). Furthermore, for the cases of 1:4,000 and 1:8,000, LD+LZ overcome the performance of the features used separately. In the case of PE+LZ, it is observed that F_1 mostly remains the same, or improves only slightly in some cases. Finally, analyzing the behavior of the combination of all the features together, it can be stated that F_1 improved in all cases.

Table 5 shows, in a more global way, two key and complementary results. In the first place, that LD is the feature that has the best individual performance. Secondly, although the features PE and LZ individually improve the results for DNN classifier, their contributions have more impact when combined. For this reason, it can be said that the novel features presented in this work provide complementary information.

In order to evaluate the statistical significance of the results, the Friedman test for F_1 was performed, resulting in a p-value of 2.5748E-05 ($\alpha = 0.01$), which indicates that there is a statistically significant difference between the scores. Then, the Nemenyi post-hoc test for F_1 was performed. This statistical analysis clearly indicates that the results obtained for LD and the combination LD+PE+LZ are the best features, in comparison to SF, LZ and PE alone. The post-hoc test showed that there are no statistically significant difference between LD and LD+PE+LZ, as it also showed that there are no statistically significant difference between LZ, PE and SF. Thus, the difference between these two groups of features is statistically significant. Furthermore, due to the fact that there were very few positive samples in the test partitions of the highest imbalances, we have repeated the experiment 10 times with different samplings of positives in the case of LD versus SF with DNN for imbalance 1:10,000. A median F_1 of 66.67% and 30.95% were obtained, for LD and SF respectively. A Wilcoxon signed-rank test test was applied to these 40 test partitions and a $p < 6.2028\text{E-}05$ was obtained.

An interesting point to further discuss is why PE and LZ individually have not shown a robust behavior

for increasing imbalances. However, when combined with LD, it has been found that those actually help improving the robustness to imbalance. This behavior suggests that these features can capture useful information from the mature, but due to its short length it is not possible to obtain values discriminative enough, by themselves, separately. However, they are more discriminative when combined with LD, because this feature does not depend on the length of the sequence itself, but on the distance to the whole reference set, as explained before. For this reason, when all the features are combined, a predominance of LD over PE and LZ is observed, although the inclusion of the latter continues to provide some discriminative information. For example, for imbalance 1:2,000, the baseline F_1 provided by SF is 49.55%, LD improves it up to 62.22% but PE and LZ are just slightly better than SF. Thus, the 65.34% of ALL is clearly dominated by LD. On the other hand, the best results of the Levenshtein distance feature can be explained based to the fact that this feature is calculated according to an external/outside set of pre-miRNAs. Instead, permutation entropy and Lempel-Ziv complexity are individual features, calculated with information within each sequence by itself. LD allows having a more accurate measure and representative sense of belonging to the positive class, since LD is a distance to a reference set of miRNAs. From another point of view, this suggests that the mature contains certain syntactic structures that guide its functioning, thus avoiding any random mutation to modify it. Therefore, by combining the information of the median distance of a candidate (LD), together with the information of its randomness (PE) and its complexity (LZ), we are restricting the number of candidate sequences just to the possible combinations of nucleotides that can allow small changes, with a defined complexity.

5 Conclusions

In the prediction of novel pre-miRNAs a large number of structural features have been proposed in order to improve the efficiency in the separation of the positive and negative classes. However, the detained performance is highly dependent on the imbalance, generating a large number of false positives. In this work, a set of new features based on the sequence information of the mature miRNA was proposed, which improve the performance independently of the classifier, decreasing the number of false positives for high imbalances. The results showed that the incorporation of the proposed measures in the mature miRNA provides a high discriminative power. Especially, the proposed Levenshtein distance has shown to have the best performance for all the imbalances. In addition, the proposed features based in permutation entropy and Lempel-Ziv complexity showed the best performances in high imbalances when combined with Levenshtein distance. The best results of the Levenshtein distance can be explained because it is a measure to a reference set of miRNAs, which allows measuring more accurately the belonging of any sequence to the positive class. This feature has provided very high precision to the classifiers evaluated, which is one of the most important contributions of our work, because most available algorithms have a very large rate of false positives. Moreover, it has shown robustness to the imbalance, improving predictions even in large imbalance scenarios. In a future work it would be interesting to introduce the probability of mutation of each nucleotide as different penalties in the Levenshtein distance. Another important conclusion of this study is that, although for all classifiers the inclusion of the new features improved their performance, the deep neural networks was the best one to relate the structural and sequence information of each pre-miRNA.

Funding

This work was supported by Universidad Nacional del Litoral (CAI+D 2016 082) and Agencia Nacional de Promocion Científica y Tecnológica (PICT 2014 2627).

References

Baker, M. (2010). MicroRNA profiling: separating signal from noise. *Nature Methods*, **7**(9), 687–692.

Bandt, C. and Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, **88**(17), 174102.

Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *cell*, **116**(2), 281–297.

Bartel, D. P. (2009). MicroRNAs: target recognition and regulatory functions. *cell*, **136**(2), 215–233.

Bartel, D. P. (2018). Metazoan microRNAs. *Cell*, **173**(1), 20–51.

Batuwita, R. and Palade, V. (2009). micropred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, **25**(8), 989–995.

Billoud, B. *et al.* (2013). Computational prediction and experimental validation of microRNAs in the brown alga Ectocarpus siliculosus. *Nucleic Acids Research*, **42**(1), 417–429.

Bortolomeazzi, M. *et al.* (2017). A survey of software tools for microRNA discovery and characterization using RNA-seq. *Briefings in Bioinformatics*, **20**(3), 918–930.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Brennecke, J. *et al.* (2005). Principles of microRNA–target recognition. *PLoS biology*, **3**(3), e85.

Brudno, M. *et al.* (2003). Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, **19**(suppl_1), i54–i62.

Bugnon, L. *et al.* (2019). Deep Neural Architectures for Highly Imbalanced Data in Bioinformatics. *IEEE Transactions on Neural Networks and Learning Systems*, **6**, 1–11.

Chen, L. *et al.* (2018). Trends in the development of miRNA bioinformatics tools. *Briefings in Bioinformatics*.

de ON Lopes, I. *et al.* (2014). The discriminant power of RNA features for pre-miRNA recognition. *BMC bioinformatics*, **15**(1), 124.

Demirci, M. D. S. *et al.* (2017). On the performance of pre-microRNA detection algorithms. *Nature communications*, **8**(1), 330.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, **7**(Jan), 1–30.

Ding, J. *et al.* (2010). MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features. *BMC bioinformatics*, **11**(11), S11.

Dong, H., , *et al.* (2013). MicroRNA: function, detection, and bioanalysis. *Chemical reviews*, **113**(8), 6207–6233.

Fischer, A. and Igel, C. (2012). An introduction to restricted boltzmann machines. In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 14–36, Berlin, Heidelberg. Springer Berlin Heidelberg.

Friedländer, M. R. *et al.* (2011). miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. *Nucleic Acids Research*, **40**(1), 37–52.

Friedman, R. C. *et al.* (2009). Most mammalian mRNAs are conserved targets of microrna. *Genome research*, **19**(1), 92–105.

Gudyś, A. *et al.* (2013). HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC bioinformatics*, **14**(1), 83.

Hertel, J. and Stadler, P. F. (2006). Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, **22**(14), e197–e202.

Jiang, P. *et al.* (2007). MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic acids research*, **35**(suppl_2), W339–W344.

Kozomara, A. *et al.* (2019). miRBase: from microRNA sequences to function. *Nucleic acids research*, **47**(D1), D155–D162.

Kozomara, A. and Griffiths-Jones, S. (2010). miRBase: integrating microRNA annotation and deep-sequencing data. *Nucleic acids research*, **39**(suppl_1), D152–D157.

Lassmann, T. and Sonnhammer, E. L. (2005). Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC bioinformatics*, **6**(1), 298.

Lempel, A. and Ziv, J. (1976). On the complexity of finite sequences. *IEEE Transactions on information theory*, **22**(1), 75–81.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Lewis, B. P. *et al.* (2003). Prediction of mammalian microRNA targets. *Cell*, **115**(7), 787–798.

Lewis, B. P. *et al.* (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *cell*, **120**(1), 15–20.

Li, L. *et al.* (2010). Computational approaches for microRNA studies: a review. *Mammalian Genome*, **21**(1-2), 1–12.

Liu, B. (2017). BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in Bioinformatics*, **20**(4), 1280–1294.

Lorenz, R. *et al.* (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, **6**(1), 26.

Mathelier, A. and Carbone, A. (2010). MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, **26**(18), 2226–2234.

Morgado, L. and Johannes, F. (2017). Computational tools for plant small RNA detection and categorization. *Briefings in Bioinformatics*, **20**(4), 1181–1192.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, **48**(3), 443–453.

Ng, K. L. S. and Mishra, S. K. (2007). De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, **23**(11), 1321–1330.

Polyanovsky, V. O. *et al.* (2011). Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for molecular biology*, **6**(1), 25.

Pritchard, C. C. *et al.* (2012). MicroRNA profiling: approaches and considerations. *Nature Reviews Genetics*, **13**(5), 358.

Shannon, C. (2001). A mathematical theory of communication. sigmobile mob comput commun rev 5 (1): 3–55.

Shukla, V. *et al.* (2017). A compilation of web-based research tools for miRNA analysis. *Briefings in functional genomics*, **16**(5), 249–273.

Stegmayer, G. *et al.* (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*, page bby037.

Takahashi, R.-u. *et al.* (2015). Loss of microRNA-27b contributes to breast cancer stem cell generation by activating enpp1. *Nature communications*, **6**, 7318.

Tseng, K.-C. *et al.* (2017). microRPM: a microRNA prediction model based only on plant small RNA sequencing data. *Bioinformatics*, **34**(7), 1108–1115.

Vitsios, D. M. *et al.* (2017). Mirnovo: genome-free prediction of microRNAs from small RNA sequencing data and single-cells using decision forests. *Nucleic Acids Research*, **45**(21), e177–e177.

Webb, A. (2002). *Statistical pattern recognition*. Wiley Press.

Wheeler, B. M. *et al.* (2009). The deep evolution of metazoan microRNAs. *Evolution & development*, **11**(1), 50–68.

Xue, C. *et al.* (2005). Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, **6**(1), 310.

Yones, C. *et al.* (2015). miRNAfe: a comprehensive tool for feature extraction in microRNA prediction. *Biosystems*, **138**, 1–5.

Yones, C. *et al.* (2018). Genome-wide pre-miRNA discovery from few labeled examples. *Bioinformatics*, **34**(4), 541–549.

Yousef, M. *et al.* (2006). Combining multi-species genomic data for microRNA identification using a naive bayes classifier. *Bioinformatics*, **22**(11), 1325–1334.

Ziv, J. and Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, **24**(5), 530–536.

Zou, Q. *et al.* (2014). miRClassify: an advanced web server for miRNA family classification and annotation. *Computers in biology and medicine*, **45**, 157–160.

Zytnicki, M. *et al.* (2008). Darn! a weighted constraint solver for RNA motif localization. *Constraints*, **13**(1-2), 91–109.

Supplementary Material
Complexity measures of the mature miRNA for improving pre-miRNAs prediction

J. Raad, G. Stegmayer and D. H. Milone

Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Santa Fe, Argentina.

1 List of standard features

- Content of guanine and cytosine $C + G_{content}$ (Hertel and Stadler, 2006):

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

  where G, C, A and U represents the count of each base found in the sequence.

- Minimum free energy (MFE) (Zuker and Stiegler, 1981): the estimated energy that one sequence frees when folded into the most stable secondary structure.

- Normalized minimum free energy of folding (dG) (Hofacker, 2003):

  $$dG = \frac{MFE}{G + C + U + A}$$

- MFE Index 1 ($MFEI_1$) (Hofacker, 2003):

  $$MFEI_1 = \frac{MFE}{C + G_{content}}$$

- MFE Index 2 ($MFEI_2$) (Hofacker, 2003):

  $$MFEI_2 = \frac{MFE}{N_{stems}}$$

  where N_{stems} is the number of stems in the secondary structure.

- The normalized Shannon entropy (dQ) (Freyhult *et al.*, 2005):

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

  where p_{ij} is the probability that the nucleotide i forms a pair with the nucleotide j and l is the sequence length (McCaskill, 1990).

- The second (the Fielder) eigenvalue (dF): An RNA secondary structure S, can be represented as a tree-graph G, where vertices represent loops, and edges represent stems. The Laplacian matrix $L(G)$ is a mathematical representation of a tree-graph G. The second eigenvalue (dF) of $L(G)$ measures the compactness of a tree-graph.

- Z-score of adjusted base pair distance (zD): In order to calculate the normalized variants for dD, a number of random sequences were generated for each original sequence in the dataset. For each generated sequence, the stability is measured with the z-score (Bonnet *et al.*, 2004), defined as

  $$zD = \frac{dD - \mu dD}{\sigma_{dD}},$$

  and

  $$dD = \frac{1}{l} \sum_{i<j} p_{ij}(1 - p_{ij})$$

  where dD is the adjusted base pair distance (Freyhult *et al.*, 2005), μdD is the mean and σ_{dD} is the standard deviation of the randomly generated population of sequences.

- Normalized Ensemble Free Energy ($NEFE$) (Hofacker, 2003): The probability of the structure S_α is given by $P(S_\alpha) = \frac{\exp^{-E_\alpha/RT}}{Z}$ where $Z = \sum_{S_\alpha \in S} \exp^{-E_\alpha/RT}$, E_α is the free energy of S_α, $R = 8.31451 Jmol^{-1}K^{-1}$ is the molar gas constant, T is the temperature taken as 310.15K and L the sequence length. Thus,

  $$NEFE = \frac{-RT \ln(Z)}{L}$$

- The structural diversity (base pair distance) ($Diversity$) (Hofacker, 2003):

  $$Diversity = \sum_{i<j} p_{ij}(1 - p_{ij})$$

  where p_{ij} is the probability of base i pair with base j.

- Energy difference ($Diff$) (Hofacker, 2003):

  $$Diff = \frac{|MFE - EFE|}{L}$$

- Structure entropy (dS) (Markham and Zuker, 2005):

  $$dS = 1000 \cdot \frac{\Delta H - \Delta G}{T}$$

  where T is the hybridization temperature, ΔG is the free energy at 37°C , ΔH is the enthalpy and the factor of 1000 expresses dS in cal/mol/K. In addition, dS/L is the Normalized structure entropy.

- MFE Index 3 ($MEFI_3$) (Batuwita and Palade, 2009):

  $$MFEI_3 = \frac{dG}{N_{loops}}$$

  where dG is normalized minimum free energy, and N_{loops} is the number of loops in the secondary structure.

- MFE Index 4 ($MFEI_4$) (Batuwita and Palade, 2009):

  $$MFEI_4 = \frac{MFE}{N_{bp}}$$

  where N_{bp} is the total number of base pairs in the secondary structure.

- Normalized base pair counts (Batuwita and Palade, 2009):

  $$NBP = \frac{|X - Y|}{L},$$

  where $|X - Y|$ is the number of $(X - Y)$ base pairs in the secondary structure, with $(X - Y) \in \{(A - U), (G - C), (G - U)\}$.

- Average base pairs per stem (Batuwita and Palade, 2009):

  $$AvgBPStem = N_{bp}/N_{stems}$$

- $(X - Y)/N_{stems}$ (Batuwita and Palade, 2009), where $(X - Y) = |X - Y|/N_{bp}$.

- tri_A, tri_U, tri_G, and tri_C (Gudyś *et al.*, 2013): frequencies of secondary structure triplets composed of three adjacent nucleotides and the middle nucleotide A(((, U(((, G(((, and C(((.

- Translation numeric (Gudyś *et al.*, 2013): the maximal length of the amino acid string without stop codons found in three reading frames

- Loops numeric (Gudyś *et al.*, 2013): the cumulative size of internal loops found in the secondary structure.

- Dustmasker numeric (Morgulis *et al.*, 2006): a percentage of low complexity regions detected in the sequence using Dustmasker.

2 Supplementary tables

Imbalance	**SF**				**SF+LD**				**SF+PE**				**SF+LZ**			
ratio	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1:500	89.38	11.26	98.61	20.00	90.00	11.87	98.68	**20.97**	89.38	11.33	98.62	20.10	90.00	11.39	98.62	20.22
1:1,000	87.50	5.91	98.60	11.06	88.75	6.37	98.69	**11.87**	88.75	6.04	98.61	11.30	87.50	5.93	98.61	11.10
1:2,000	95.00	3.47	98.69	6.70	95.00	3.67	98.76	**7.06**	95.00	3.45	98.68	6.65	95.00	3.45	98.68	6.66
1:4,000	95.00	2.98	99.22	5.78	95.00	3.45	99.34	**6.65**	95.00	2.99	99.22	5.80	95.00	3.00	99.23	5.81
1:6,000	100.00	2.23	99.29	4.35	100.00	2.93	99.47	**5.68**	100.00	2.34	99.33	4.56	100.00	2.27	99.30	4.43
1:8,000	100.00	2.10	99.53	4.12	100.00	3.27	99.70	**6.33**	100.00	2.21	99.54	4.33	100.00	2.23	99.55	4.36
1:10,000	87.50	2.10	99.56	4.08	87.50	3.17	99.69	**6.10**	87.50	2.21	99.59	4.30	87.50	2.18	99.59	4.24

Table S1: Classification results for each new feature and standard features with NB.

Imbalance	**SF**				**SF+LD**				**SF+PE**				**SF+LZ**			
ratio	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1:500	50.62	86.48	99.98	63.3	52.50	86.35	99.98	64.60	52.50	85.87	99.98	64.55	55.00	82.99	99.98	**65.41**
1:1,000	36.25	87.08	100.00	50.99	37.50	88.75	100.00	**52.62**	32.50	76.88	99.99	45.30	35.00	85.00	100.00	49.23
1:2,000	17.50	70.83	100.00	27.88	25.00	83.33	100.00	37.18	20.00	72.92	100.00	31.14	25.00	83.33	100.00	**38.00**
1:4,000	20.00	50.00	100.00	26.59	15.00	41.67	100.00	20.83	25.00	54.17	100.00	**33.93**	15.00	37.50	100.00	19.44
1:6,000	0.00	0.00	100.00	0.00	8.33	25.00	100.00	**12.50**	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00
1:8,000	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00
1:10,000	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	0.00

Table S2: Classification results for each new feature and standard features with RF.

Imbalance	**SF**				**SF+LD**				**SF+PE**				**SF+LZ**			
ratio	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1:500	50.62	55.66	99.92	52.71	55.62	60.41	99.92	**57.39**	50.62	56.88	99.92	53.13	50.62	56.48	99.92	53.11
1:1,000	40.00	43.08	99.95	41.05	42.50	47.84	99.95	**44.70**	40.00	44.14	99.95	41.39	41.25	44.83	99.94	41.87
1:2,000	27.50	66.67	99.99	38.32	37.50	68.96	99.99	**47.76**	25.00	66.67	99.99	35.58	32.50	57.86	99.99	41.37
1:4,000	35.00	76.67	100.00	45.12	35.00	83.33	100.00	**47.62**	35.00	75.00	99.99	44.21	35.00	76.67	100.00	45.12
1:6,000	16.67	50.00	100.00	25.00	33.33	41.67	100.00	**36.67**	16.67	37.50	100.00	22.50	16.67	37.50	100.00	22.50
1:8,000	12.50	25.00	100.00	**16.67**	12.50	25.00	100.00	**16.67**	12.50	25.00	100.00	**16.67**	12.50	25.00	100.00	**16.67**
1:10,000	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

Table S3: Classification results for each new feature and standard features with KNN.

Imbalance	**SF**				**SF+LD**				**SF+PE**				**SF+LZ**			
ratio	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1	s^+	p	s^-	F_1
1:500	65.00	79.65	99.96	69.50	68.12	77.88	99.96	**71.44**	66.88	77.48	99.95	70.65	65.00	77.53	99.96	69.34
1:1,000	42.50	68.37	99.97	48.81	37.50	81.04	99.99	**49.33**	35.00	71.25	99.98	45.33	36.25	71.64	99.98	46.05
1:2,000	47.50	62.20	99.98	49.55	62.50	68.86	99.98	**62.22**	65.00	56.61	99.95	54.82	52.50	55.13	99.97	48.29
1:4,000	55.00	67.92	99.99	60.28	75.00	70.83	99.99	**70.78**	60.00	62.50	99.99	61.11	75.00	58.48	99.99	64.81
1:6,000	41.67	23.21	99.99	29.29	58.33	69.17	100.00	**57.92**	33.33	22.50	99.99	26.79	50.00	31.25	99.99	38.10
1:8,000	37.50	29.17	99.99	32.50	87.50	70.83	100.00	**77.50**	50.00	19.64	99.99	27.36	50.00	25.00	99.99	32.50
1:10,000	37.50	32.14	99.99	27.78	75.00	91.75	100.00	**78.33**	37.50	45.83	100.00	39.17	37.50	58.33	100.00	43.33

Table S4: Classification results for each new feature and standard features with DNN.

3 Supplementary figure

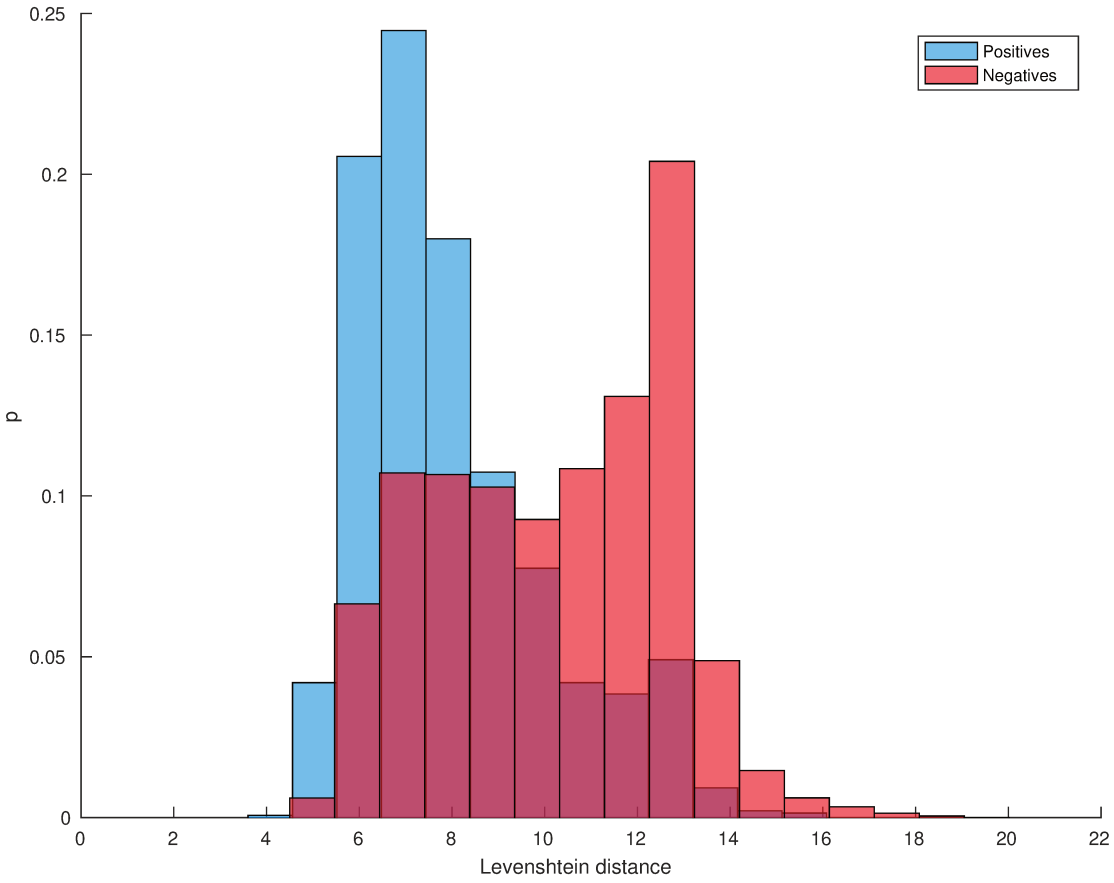

Figure S1: Histogram of LD feature for the positive and negative classes, for the complete dataset (1,406 positives and 81,228 negatives).

References

Batuwita, R. and Palade, V. (2009). micropred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, **25**(8), 989–995.

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

Freyhult, E., Gardner, P. P., and Moulton, V. (2005). A comparison of RNA folding measures. *BMC Bioinformatics*, **6**(1), 241.

Gudyś, A. *et al.* (2013). HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC Bioinformatics*, **14**(1), 83.

Hertel, J. and Stadler, P. F. (2006). Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, **22**(14), e197–e202.

Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Research*, **31**(13), 3429–3431.

Markham, N. R. and Zuker, M. (2005). DINAmelt web server for nucleic acid melting prediction. *Nucleic Acids Research*, **33**(suppl_2), W577–W581.

McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers: Original Research on Biomolecules*, **29**(6-7), 1105–1119.

Morgulis, A., Gertz, E. M., Schäffer, A. A., and Agarwala, R. (2006). A fast and symmetric DUST implementation to mask low-complexity DNA sequences. *Journal of Computational Biology*, **13**(5), 1028–1040.

Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9**(1), 133–148.

miRe2e: a full end-to-end deep model based on Transformers for prediction of pre-miRNAs from raw genome-wide data

miRe2e: a full end-to-end deep model based on Transformers for prediction of pre-miRNAs from raw genome-wide data

J. Raad, L.A. Bugnon, D. H. Milone and G. Stegmayer

Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Santa Fe, Argentina.

Abstract

Motivation: MicroRNAs (miRNAs) are small RNA sequences with key roles in the regulation of gene expression at post-transcriptional level in different species. Accurate prediction of novel miRNAs is needed due to their importance in many biological processes and their associations with complicated diseases in humans. Many machine learning approaches were proposed in the last decade for this purpose, but requiring handcrafted features extraction in order to identify possible de novo miRNAs. More recently, the emergence of deep learning has allowed the automatic feature extraction, learning relevant representations by themselves. However, the state-of-art deep models require complex pre-processing of the input sequences and prediction of their secondary structure in order to reach an acceptable performance. **Results:** In this work we present miRe2e, the first full end-to-end deep learning model for pre-miRNA prediction. This model is based on Transformers, a neural architecture that uses attention mechanisms to infer global dependencies between inputs and outputs. It is capable of receiving the raw genome-wide data as input, without any pre-processing nor feature engineering. After a training stage with known pre-miRNAs, hairpin and non-harpin sequences, it can identify all the pre-miRNA sequences within a genome. The model has been validated through several experimental setups using the human genome, and it was compared with state-of-the-art algorithms obtaining 10 times better performance.
Availability: Webdemo available at https://sinc.unl.edu.ar/web-demo/miRe2e/ and source code available for download at https://github.com/sinc-lab/miRe2e
Contact: jraad@sinc.unl.edu.ar.
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

MicroRNAs (miRNAs) can regulate genes, determine the genetic expression of cells, influence the state of the tissues and promote or inhibit certain diseases and infections (Bartel, 2004). The discovery of new miRNAs and their function is necessary for better understanding their roles in genes regulation. The precursors of miRNAs (pre-miRNAs) generated during biogenesis have a well-known RNA secondary structure, which has allowed the development of computational algorithms for their identification. The pre-miRNAs typically exhibit a stem-loop structure, which are also known as hairpin, with few internal loops or asymmetric bulges. However, a very large amount of hairpin-like structures can be found in a genome, thus the discovery of truly pre-miRNAs remains a challenge.

For the prediction of pre-miRNAs, there is a large number of pipelines that use genomics data as input for building a binary classifier based on machine learning (ML) (Stegmayer *et al.*, 2018; Bugnon *et al.*, 2020). All of them need an intensive pre-processing of the raw genome: set a window length, go through the genome and cut it into fixed sequences, calculate the corresponding secondary structure, check that it forms a hairpin and discard those sequences that do not (named flats). Then, a large number of handcrafted features are extracted from the harpins, such as the number of loops or the minimum free energy when folding the secondary structure (MFE), among many others (de O. N. Lopes *et al.*, 2014; Yones *et al.*, 2015; Raad *et al.*, 2020). The MFE has proved to be one of the most important

features for distinguishing pre-miRNAs (Bartel, 2004). This feature extraction step is highly dependent on the manual selection of many parameters, and these human decisions in pre-processing can have an impact in the prediction afterwards. The ML classifiers are then trained to learn those features from positive (well-known pre-miRNAs deposited in miRBase) and negative class samples, for the discovery of new pre-miRNAs in non-coding and non-repetitive regions of any genome.

In several bioinformatics domains, the big challenge today is the development of ML methods without requiring any pre-processing of the input, that is, a so-called end-to-end model (Trieu *et al.*, 2020; Tsubaki *et al.*, 2018; Chaabane *et al.*, 2019). In the scenario of genome-wide pre-miRNAs prediction, such a method should be able to be trained only with raw RNA sequences (no features), and then be able to receive the raw genome of any species without any features extraction nor calculation of secondary structure, to identify hairpin-like pieces of RNA highly likely to be novel pre-miRNAs. However, since in such a scenario it is not possible to previously discard those sequences that do not fold as hairpins (the flats), it is necessary to incorporate all them into the training. Precisely for avoiding any feature engineering step, the emergence of deep learning (DL) has produced meaningful improvements in the field of automatic representation for computer vision, speech recognition and many other application domains (LeCun *et al.*, 2015). Deep models can automatically extract relevant features by themselves, directly from raw data, and those are considered today the best paradigm of ML for most classification tasks (Jurtz *et al.*, 2017; Bengio *et al.*, 2013). DL has already been used for small-RNA feature extraction, identification and classification (Zheng *et al.*, 2019; Amin *et al.*, 2019; Zeng *et al.*, 2016). In addition, DL can detect motifs in a set of homologous sequences, which are then the key for distinguishing among different types of protein families or predict its structure (Seo *et al.*, 2018; Senior *et al.*, 2020). In (Eraslan *et al.*, 2019) authors analyze gaps and challenges for DL in genomics, mentioning the need for more DL-based tools capable of handling the real genome-wide scenario with full end-to-end models, without requiring any type of handcrafted pre-processing.

In this line of work, very recently a model based on convolutional neural networks (CNN), named deepMir, has been proposed for classification of miRNA families (Tang and Sun, 2019). Differently from most binary classification tools, the focus here is on classifying input sequences into different miRNA families for more detailed function annotation. It receives as input only RNA sequences, using a one-hot-encoding scheme to convert a RNA sequence of $1 \times N$ nt into an $4 \times N$ matrix to feed the network, coding this way the 4 nucleotides types in the sequence. The CNN model contains two convolutional layers, followed by max pooling layers and three fully connected layers with dropout. The model is trained with pre-miRNAs from Rfam and mature miRNAs from miRBase. In (Bugnon *et al.*, 2020) it was shown that the performance of deepMir was below those deep models that use also the predicted secondary structure as input, such as deepMiRGene (Park *et al.*, 2017). However, deepMir is an important step towards models fully trainable from raw genomic sequences and a starting point for achieving end-to-end models, with the potential of outperforming other approaches thanks to the capability of learning the features automatically. Nevertheless, it should be noted that deepMir has not been designed nor tested for discovery of novel pre-miRNAs in a genome-wide scenario. Moreover, pure CNNs have shown some limitations for the analysis of sequences, due to the locality of its convolutions and the loss of long-term dependencies, requiring the stacking of several layers (Vaswani *et al.*, 2017).

As an alternative to improve DL models in the automatic extraction of features, the Transformers have appeared very recently, coming from the natural language processing domain (Devlin *et al.*, 2018; Vaswani *et al.*, 2017). Transformers are deep networks with self-attention mechanisms in each layer, which allows obtaining several improvements with respect to recurrent and convolutional models (Dosovitskiy *et al.*, 2020). On the one hand, the information flow is parallelized, instead of being done sequentially as in recurrent networks. On the other hand, unlike convolutional networks that work with a local vision and require many layers to obtain a global vision, the attention mechanisms allow the analysis of longer sequences without losing context information, and maintaining a global vision of the input in each layer, due to their point-to-point connections (Vaswani *et al.*, 2017). These characteristics of the Transformers allow learning relationships between all nucleotides within a hairpin like sequence, thus being able to model its secondary structure. This way, it is possible to develop a deep learning model capable of, only from the raw RNA sequence of a pre-miRNA, extracting information from its secondary structure without any data preprocessing and engineered feature extraction.

In this work, we propose miRe2e, a full end-to-end deep learning model for pre-miRNA prediction

based on Transformers and attention mechanisms. It is capable of receiving as input the sequences of raw genome-wide data, without any pre-processing. After a training step with known and unlabeled sequences, it can identify pre-miRNA sequences within a genome. This model automatically learns the intrinsic structural characteristics of precursors of miRNAs from the raw data, without any feature engineering. The proposal has been tested with several experimental setups with the human genome, and compared with state-of-the-art algorithms.

2 Full end-to-end deep learning model

The miRe2e is a full end-to-end deep learning model based on Transformers. A Transformer is a neural model architecture that relies on attention mechanisms to infer global dependencies between input and output. Each Transformer is made up of layers of attention mechanisms and feedforward networks (Vaswani *et al.*, 2017). The attention mechanisms aim at finding relationships between each pair of elements within a sequence (that is, between nucleotides in a genomic sequence) (Bahdanau *et al.*, 2015). To do this, a dot product is calculated between each pair of elements, thus obtaining a score matrix. Then, softmax is applied to each row of the matrix, obtaining the weights associated with the product of each nucleotide with the rest of the nucleotides in the sequence. For obtaining the new vector associated with each nucleotide, a dot product is made between its weight vector, that is the corresponding row of the score matrix, and the full sequence. Finally, an output sequence of the same dimension as the input sequence is obtained, but where each nucleotide is weighted by its importance in its context. In particular, several sets of these weights can be learnt to capture different relationships in the sequence, giving rise to the so-called multihead attention (Vaswani *et al.*, 2017). In this case, instead of using a single large weights matrix, each nucleotide is projected in parallel into a set of matrices of less dimension, which are called heads. The output of each head is concatenated into a single vector, which is projected to obtain a single output. This allows the model to obtain information from different subspaces, thus achieving a better representation for each nucleotide position in the sequence.

Since the miRe2e is designed to work on genome-wide without any pre-processing, the input sequences are obtained through a scan and cut of the raw genome with overlapped windows, with a window of length L and step s. Then, each sequence is represented as a $L \times 4$ one-hot-encoding tensor, where each column represents one of the four possible nucleotides (A, C, G, U) at each position. The miRe2e processes this input with 3 internal deep models, as depicted in Figure 1: the Structure Prediction (A), the MFE Estimation (B) and the pre-miRNA Classifier (C). The figure shows the complete miRe2e model, where the input/outputs of each model are shown with numbers and the details of the neural architecture of each model are shown immediately below. The Structure Prediction model allows obtaining the secondary structure from a RNA input sequence. The MFE Estimation model calculates the MFE from an input RNA sequence and its corresponding secondary structure. Finally, the last deep model performs the pre-miRNA classification.

The Structure Prediction model (Figure 1A) learns to estimate the secondary structure from a RNA sequence. Here the one-hot-encoding tensor $\boxed{1}$ enters a CNN of three stages, each one with identity blocks. Each one of these identity blocks is made up of two activation functions, two batch normalization layers, two one-dimensional convolutional layers of length L, and w_A filters with identity shortcut connections (He *et al.*, 2016). The main function of this part of the model is to automatically extract motifs from the input sequence and increase the number of features to allow a fast processing in attention layers (Vaswani *et al.*, 2017). At the output of the CNN, the positional encoding signal is added to each embedding (Vaswani *et al.*, 2017). Then, there is a stack of six Transformer encoders. In this part of the model, each encoder layer is composed of w_A input features, h_A heads, and n_A neurons in the hidden layers of each feedforward network, where the number of hidden neurons is set to $n_A = 4w_A$ as suggested in (Vaswani *et al.*, 2017). The function of this encoder is, through its attention mechanisms, to model the contact matrix of each nucleotide in the input sequence, thus being able to estimate its secondary structure. Finally, after the encoder there is a 3-layer multilayer perceptron (MLP), ELU activation functions in the hidden layers and hyperbolic tangent functions at the output are used. Since in Transformer encoders the output has the same dimension as the input

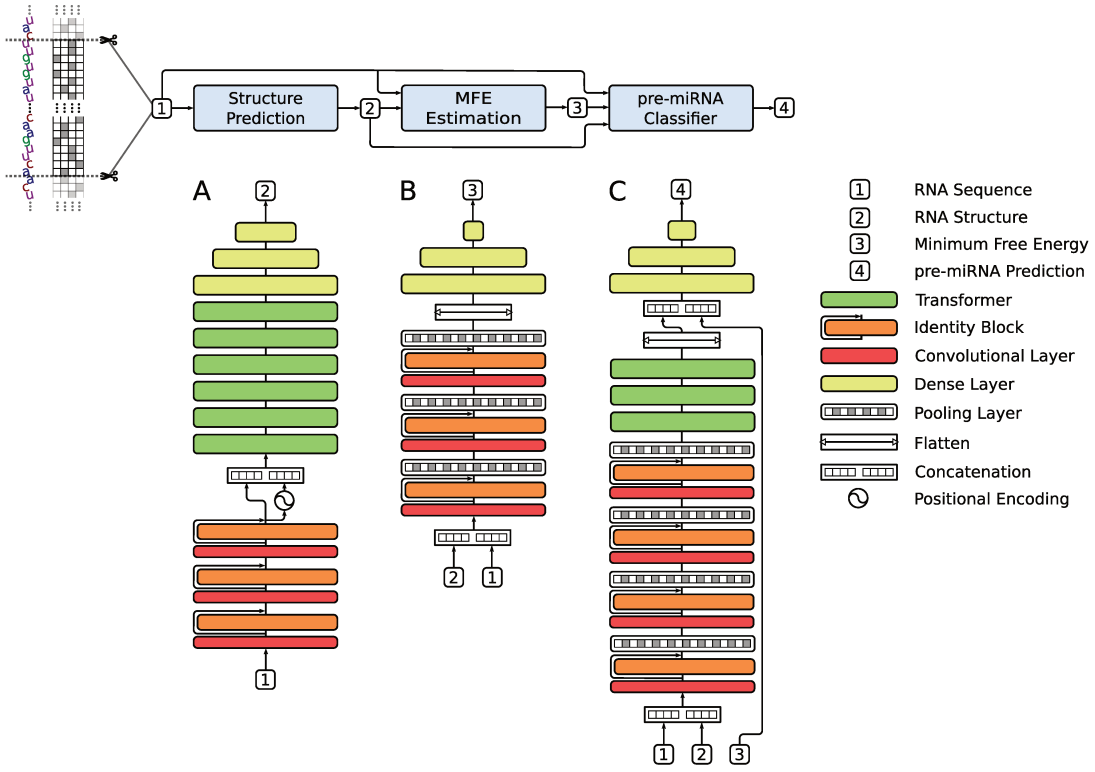

Figure 1: Schematic representation of the complete miRe2e: full end-to-end architecture for pre-miRNA prediction in genome-wide data. The details of the architecture of each model are shown below. (A) The input RNA sequence [1] enters the Structure Prediction model, which outputs the RNA structure [2]. (B) The MFE Estimation model receives [1] and [2] and calculates the Minimum Free Energy [3]. (C) The pre-miRNA Classifier model receives [1], [2] and [3] and provides the pre-miRNA prediction [4].

$(L \times w_A)$, and the MLP is applied to each sample of the input tensor without flattening, a reduction in the dimension of features from w_A to 1 is obtained, generating a tensor of $L \times 1$ at the output [2]. To avoid the bias towards the non pre-miRNA sequences due to the high class-imbalance, class oversampling was done, where each training batch is constructed with the same number of samples from the minority class (actual pre-miRNAs) and the majority class. To do this, the minority class was sampled with replacement. Finally for this model, the mean squared error (MSE) loss function was used for training, which is calculated between the estimated $L \times 1$ output tensor and the reference secondary structure. This was represented with 0, 1 and -1 for unmatched nucleotides, matches in the 5' strand and matches in the 3' strand, respectively. Thus, it is possible to encode the two strands of each hairpin into a single real vector (Yones *et al.*, 2021).

The second model (Figure 1B) aims to estimate the MFE from the input sequence and its secondary structure. It receives [1] and [2], concatenates them and obtaining a $5 \times L$ tensor with the fifth row being the secondary structure predicted for the input sequence. The model is made up of a 3-stage CNN, each one composed of an identity block and a stacked pooling layer. Due to each pooling layer, after each stage the length of the input tensor is reduced by a factor of 2. At each identity block, the one-dimensional convolutional layers are formed by w_B filters and a $L/(2^N)$ length, where N is the stage number. Then, after a flatten layer, there is a 3-layer MLP where each of these layers has batch normalization and ELU activation functions. MSE loss was used for training, as the error function

between each predicted output value and its reference MFE value. The output of this CNN is the estimated MFE [3] of the sequence.

The pre-miRNA Classifier model (Figure 1C) classifies the input sequence [1], with its secondary structure [2] and the estimated MFE [3]. This model has a 4-stage CNN, each made up of three identity blocks with w_C filters and a stacked pooling layer. Then, there is a stack of three Transformers encoders. Each encoder layer has w_C input features, h_C heads and n_C neurons in the hidden layers of each feedforward network. Its function is to encode the sequential information of the input, thus modeling the dependency between each nucleotide in a global way. After the encoder, the $w_C \times L/16$ output tensor is flattened and concatenated with the output of the MFE model [3]. After that, it goes to a 4-layer MLP, hidden ELU activation functions, batch normalization and dropout. Finally, a softmax layer at the output predicts the corresponding class for the input sequence [4]. Since miRe2e is composed of three models in cascade, a 3-stage training was carried out, where the output of each model was the input of the next one. More details about the miRe2e hyperparameters and training can be found in the Supplementary Material.

3 Materials and Methods

3.1 Data

Genome-wide data of *Homo sapiens*[1] was used in all the experiments (Bugnon *et al.*, 2019). For training the first model (secondary structure prediction), all the metazoan pre-miRNAs (23,178), excluding *H. sapiens*, obtained from mirBase v.22[2] were used, and 2,000,000 pseudo-hairpins were extracted from the genome with HextractoR (Yones *et al.*, 2020). The target data for each input sample is its corresponding secondary structure predicted with RNAfold (Hofacker, 2003), at a temperature of 37C.

For training the deep model that estimates the MFE, the secondary structure predicted by the first model and its respective input RNA sequence are required. The desired output here was the RNAfold predicted MFE value normalized by sequence length. For this model, 23,178 metazoan pre-miRNAs (excluding *H. sapiens*) were used. In addition, 48,000 pseudo-hairpins obtained with HextractoR and 48,000 sequences that did not form hairpins (flats) were randomly extracted from the genome. For testing the complete model, the input sequences are obtained through a scan and cut of each chromosome with overlapped windows (length 100 nt, step 20 nt).

3.2 Performance evaluation

The methods performance is reported with standard recall or sensitivity (s^+), precision (p), and F_1 evaluation metrics,

$$s^+ = \frac{TP}{TP + FN}, \qquad p = \frac{TP}{TP + FP}, \qquad F_1 = 2\ \frac{s^+\ p}{s^+ + p},$$

where TP, FP and FN are the number of true positives, false positives and false negatives, respectively. The recall measures how good a classification method is for recognizing the TPs of the task. The precision measures the relation between TPs and FPs. In a realistic scenario for practical applications, precision is very important for datasets with high class imbalance, because FPs can be many more than the TPs. Thus, considering the characteristics of the classification task under study, it is important to take into account both sensitivity and precision. Therefore, F_1 is used as a global comparative measure. It should be noted here that in this scenario of such high class imbalance, very low values can be expected from these measures. For example, if a predictor has only 1% of FP in a dataset with $1,000$ TP and $10,000,000$ total sequences, the precision could be below 0.001. As a consequence, very low values of F_1 will be also observed. For performance evaluation and comparison with other methods,

[1] http://ftp.ensembl.org/
[2] http://www.mirbase.org/

Fold	s^+		p		F_1	
	miRe2e	deepMir	miRe2e	deepMir	miRe2e	deepMir
1	0.0130	**0.0250**	**0.0020**	0.0002	**0.0030**	0.0005
2	**0.0130**	**0.0130**	**0.0020**	0.0004	**0.0040**	0.0008
3	**0.0380**	0.0130	**0.0010**	0.0006	**0.0020**	0.0012
4	0.0130	**0.1150**	**0.0110**	0.0005	**0.0120**	0.0011
Avg.	0.0193	**0.0415**	**0.0040**	0.0004	**0.0052**	0.0009

Table 1: Performance comparison of miRe2e and deepMir for the prediction of pre-miRNAs in the chromosome 1 of *H. sapiens*.

a 4-fold stratified cross-validation strategy was used, that is, preserving the original percentage of each class on each fold.

These measures were also used to obtain precision recall curves (PRC), which is a well-known indicator for global performance of classifiers. It has been shown (Saito *et al.*, 2015) that this measure is preferred over the classical receiver operating characteristic (ROC) curve to assess binary classifiers on highly imbalanced data. When there is a large class imbalance in a dataset, a classifier can reach a good performance in terms of specificity (and sensitivity), but can perform poorly in providing good quality candidates, with a large amount of false positives. A PRC can provide a better assessment of performance because it also evaluates the fraction of true positives among the total positive predictions. The area under the precision-recall curve (AUCPR), which is a single numeric summary of the information, will also be reported as a global measure along all the possible output thresholds in the compared models.

4 Results

4.1 Comparison with state-of-the-art methods

In order to show the better generalization capability of our model, a comparison of predictions in cross-validation for the chromosome 1 of *H. sapiens* was done. Training data included all positives (156 known pre-miRNAs) in chromosome 1 and the rest of the sequences of chromosome 1 (more than 24,000,000), divided into 4-folds for training and testing. We compared the performance obtained with miRe2e for this task against the most recently proposed pre-miRNA prediction tool, deepMir (Tang and Sun, 2019), which also receives raw input sequences (that is, without preprocessing and feature extraction).

The results are shown in Table 1, which reports each fold results in the rows, and then s^+, p and F_1 for each method, respectively. Regarding s^+, both methods have good results, being deepMir slightly better on average. Instead, the precision of miRe2e is always the best one, in all folds. It is quite remarkable here that the performance of miRe2e is one order of magnitude higher than deepMir. This is precisely reflected by F_1, where miRe2e is always superior to deepMir, in all cases with one order of magnitude of difference. This is due to the fact that miRe2e can effectively model the secondary structure of the RNA sequence, and since this information is key for filtering false positives. The model can improve p without a drop in s^+, thus increasing the global F_1. It should be noted that these significant results were obtained in the context of the high class imbalance of one chromosome (156 positive versus 24,000,000 negative samples), which suggest that the performance of miRe2e in a complete genome-wide scenario can be superior to deepMir.

4.2 Prediction of human pre-miRNAs added in the future in miRBase

To further test the performance of miRe2e in a more realistic scenario, involving the prediction of novel pre-miRNAs in the future, we trained it on the human pre-miRNAs dataset from miRBase v17 (2011) and tested with the human pre-miRNAs introduced afterwards in miRBase v21 (2014). Thus,

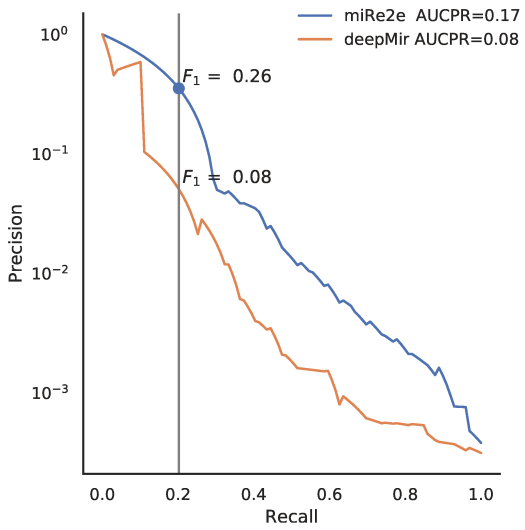

Figure 2: Precision recall curves for miRe2e and deepMir, for the prediction of human pre-miRNAs recently added in miRBase.

the training set has 1,854 positive, 87,500 negative and 787,500 flat sequences, and the test set was composed of 27 positive, 12,500 negative and 112,500 flat sequences.

The PR curves are shown in Figure 2 for both models in different colors. It can be seen that miRe2e (blue line) has reached the best results, with AUCPR = 0.17. The deepMir method (orange line) has obtained AUCPR = 0.08, a very low value and one order of magnitude less than miRe2e.

It should be noticed that, for the same recall in both methods (e.g. 0.20), while miRe2e obtains a $F_1 = 0.26$ with 11 FP, deepMir has $F_1 = 0.08$ with 113 FP (more than 10 times). This is of high importance in the application domain, where if for the same TP rate a large number of initial candidates to novel pre-miRNAs are obtained, in the order of hundreds or thousands, it will be almost impossible to validate them all experimentally in order to discover real pre-miRNAs. Thus, a smaller number of predicted and good candidates is preferred. These results show that miRe2e is effective for the prediction and discovery of new pre-miRNAs in the future.

4.3 Genome-wide discovery of pre-miRNAs in a new species

Finally, to test miRe2e in a very realistic task of discovery novel pre-miRNAs in a new species, the following experimental setup has been used. The Structure Prediction model was trained with all known metazoan pre-miRNAs excluding *H. sapiens* (23,048 sequences), and negative samples from animals (2,000,000 hairpins in total from *Anopheles gambiae*, *Drosophila melanogaster* and *Caenorhabditis elegans*) (Bugnon *et al.*, 2019). The MFE model was trained with all known metazoan pre-miRNAs excluding *H. sapiens*, 48,000 pseudo-hairpins and 48,000 flats randomly extracted from *Anopheles gambiae*, *Drosophila melanogaster* and *Caenorhabditis elegans*. The pre-miRNA Classifier model was trained with all known metazoan pre-miRNAs excluding *H. sapiens* (23,048), and negative samples from animals (1,000,000 in total from *Anopheles gambiae*, *Drosophila melanogaster* and *Caenorhabditis elegans*: 100,000 hairpins and 900,000 flats). The task was the discovery of all the pre-miRNAs in the human genome (as if it were a novel species recently discovered). Thus, for testing, all the sequences within each chromosome containing a known pre-miRNA, according to the positions described in miRBase v22, were used as the positive class, and the negatives were all the corresponding sequences from the rest of the chromosome.

The results are presented in Table 2. The first column indicates the chromosome, and the second and third column the number of positive and negative examples in that chromosome, respectively. Then the performance of each method is reported with s^+, p, F_1, AUROC and AUPRC. Finally, the last row indicates the final performance measured in the full human genome. It should be noticed the

Chr	Pos	Negatives	deepMir					miRe2e				
			s^+	p	F_1	AUROC	AUPRC	s^+	p	F_1	AUROC	AUPRC
1	156	24,895,488	0.013	0.0020	0.0035	0.7115	0.00004	0.235	0.0040	0.0079	0.9439	**0.11880**
2	116	24,213,504	0.075	0.0002	0.0003	0.7081	0.00003	0.271	0.0038	0.0075	0.9640	**0.13673**
3	96	19,826,688	0.063	0.0001	0.0002	0.7024	0.00002	0.240	0.0043	0.0085	0.9623	**0.12117**
4	62	19,015,680	0.172	0.0000	0.0001	0.7442	0.00002	0.190	0.0025	0.0050	0.9724	**0.09576**
5	76	18,149,376	0.013	1.0000	0.0263	0.6978	0.01335	0.280	0.0045	0.0089	0.9585	**0.14131**
6	71	17,080,320	0.071	0.0001	0.0002	0.7885	0.00002	0.271	0.0043	0.0084	0.9771	**0.13684**
7	82	15,931,392	0.013	0.0004	0.0008	0.7880	0.00004	0.138	0.0019	0.0038	0.9537	**0.06949**
8	90	14,512,128	0.012	0.0030	0.0048	0.7488	0.00016	0.232	0.0044	0.0086	0.9196	**0.11732**
9	88	13,836,288	0.012	0.0014	0.0025	0.6767	0.00003	0.318	0.0056	0.0109	0.9580	**0.16041**
10	69	13,375,488	0.030	0.0001	0.0003	0.7041	0.00003	0.333	0.0044	0.0086	0.9676	**0.16804**
11	102	13,504,512	0.040	0.0004	0.0007	0.8100	0.00008	0.228	0.0042	0.0082	0.9669	**0.11528**
12	80	13,326,336	0.013	0.0526	0.0206	0.7694	0.01288	0.231	0.0042	0.0082	0.9382	**0.11621**
13	40	11,433,984	0.025	0.0000	0.0001	0.7227	0.00001	0.150	0.0027	0.0053	0.9581	**0.07596**
14	99	10,702,848	0.041	0.0004	0.0009	0.7364	0.00004	0.204	0.0061	0.0119	0.9726	**0.10385**
15	71	10,199,040	0.044	0.0002	0.0005	0.6519	0.00003	0.324	0.0066	0.0129	0.9564	**0.16360**
16	82	9,031,680	0.148	0.0003	0.0007	0.6709	0.00009	0.210	0.0035	0.0069	0.9427	**0.10615**
17	110	8,325,120	0.075	0.0002	0.0004	0.6709	0.00004	0.142	0.0028	0.0054	0.9501	**0.07162**
18	35	8,036,352	0.031	0.0001	0.0003	0.6778	0.00002	0.250	0.0040	0.0080	0.9430	**0.12595**
19	143	5,861,376	0.007	0.0014	0.0024	0.8321	0.00018	0.300	0.0077	0.0150	0.9661	**0.15277**
20	48	6,438,912	0.021	0.0024	0.0043	0.7646	0.02131	0.234	0.0034	0.0067	0.9727	**0.11774**
21	33	4,669,440	0.032	0.0067	0.0111	0.7025	0.03229	0.161	0.0034	0.0067	0.9610	**0.08132**
22	46	5,861,376	0.068	0.0002	0.0003	0.6272	0.00002	0.295	0.0038	0.0075	0.9412	**0.14882**
X	118	15,599,616	0.009	0.0013	0.0023	0.7605	0.00003	0.301	0.0096	0.0186	0.9741	**0.15368**
Y	4	5,720,064	0.250	0.0001	0.0003	0.5668	0.00002	0.500	0.0003	0.0006	0.7954	**0.00010**
Full	1,917	309,547,008	0.004	0.0003	0.0006	0.7117	0.00003	0.244	0.0043	0.0085	0.9595	**0.12313**

Table 2: Performance comparison of miRe2e and deepMir for the prediction of pre-miRNAs in the genome of *H. sapiens*. Detailed measures for each chromosome (Chr) and the full genome (Full row).

very large class imbalance that exists in each chromosome. For example, in chromosome 1 there are 156 positives and more than 24 millions of negatives, that is, an imbalance ratio of about 1:160,000. Even worse, in chromosome Y there are only 4 positives and more than 5 millions of negatives, making the imbalance ratio up to 1:1,430,000. As stated in Section 3.2, in such scenarios very low values of F_1 can be expected. Note that in this case, with just 1% of FP, the F_1 drops below 0.0001, thus the global measures of AUROC and AUPRC are an important complement for the analysis of these results.

The results shown in Table 2 indicate that, in spite of the very large class imbalance existing in each chromosome, the miRe2e model has the best results in all cases. With respect to s^+, miRe2e is twice better than deepMir for all chromosomes. Regarding p, the precision is the best one, even one order of magnitude higher in most cases. In particular, for chromosome 2, the miRe2e performance in precision is 20 times better than deepMir. In the only case where deepMir has $p = 1.00$ (chromosome 5), it should be noticed however that the corresponding sensitivity is $s^+ = 0.013$ (in contrast to $s^+ = 0.280$ for miRe2e). Although at this (s^+, p) point deepMir maximizes F_1, this is achieved at the cost of a very low sensitivity. For F_1 and AUROC measures, again, miRe2e clearly outperforms deepMir in all chromosomes. Finally, regarding the best performance measure for this type of problems with very large class imbalance, AUPRC, the best result for each chromosome is indicated in bold. As it can easily be seen from the table, all best results correspond to miRe2e.

As a final comparison, not only with deep learning methods that use raw data but also with one of the best current methods that uses the predicted secondary structure of the sequences, we have made a full genome-wide experiment. Figure 3 shows the PR curves for the complete human genome (using all sequences from all chromosomes), for miRe2e (raw data), deepMir (raw data) and deepMiRGene (raw data + secondary structure). Although the last one is not a full end-to-end deep model, because it uses the secondary structure predicted by an external non-neural model (RNAfold), it provides a valid comparison with a state-of-the-art reference. In the top left of Figure 3 it can be clearly seen that the best performance is for miRe2e, with the largest difference with respect to the other methods. At the highest recalls (>0.6), miRe2e behaves equally to deepMiRGene and much better than deepMir. However, note that this part of the PR curve is of very limited practical utility, given the high number of false positives in this highly-imbalanced scenario. It should be mentioned that this high performance

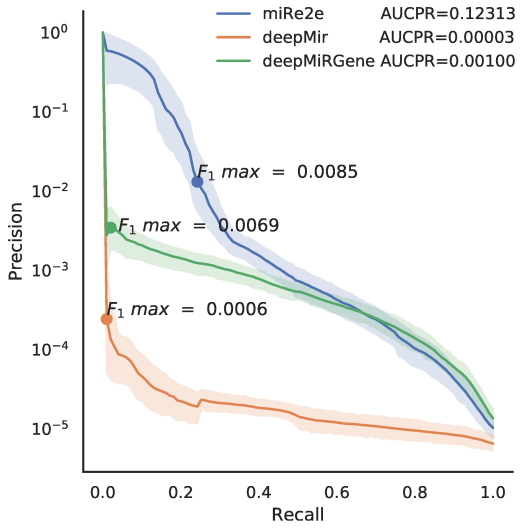

Figure 3: Precision recall curves for miRe2e, deepMir and deepMiRGene for the prediction of human pre-miRNAs in the complete genome.

for miRe2e is obtained without requiring any other information than the raw sequence. Remarkably, in this experiment, the total AUCPR for miRe2e is 0.12313, which is more than 10 times higher than the other methods.

These results indicate that miRe2e can be reliably used for the discovery of novel pre-miRNAs in a full genome, with the best possible sensitivity and precision in such a high imbalance scenario. That is, with a very low number of positive examples to learn for the discovery of new ones. This makes miRe2e the first full end-to-end deep learning model, based in Transformers, for the pre-miRNA prediction task.

5 Conclusions

In this work we have proposed miRe2e, the first full end-to-end deep learning model for pre-miRNA prediction in genome-wide data. The advantages of this model over state-of-the methods are twofold. On the one hand, it is capable of receiving raw genome-wide data, without any pre-processing or secondary structure prediction. Thus, it is possible to minimize the impact of handcrafted processes, and improve the reproducibility and replicability of results. On the other hand, miRe2e can identify all the pre-miRNA sequences within a genome with very high precision and recall. Moreover, it has shown not to be affected by the very high class imbalance that exists within a full genome between possible novel pre-miRNAs and the huge amount of negative sequences. In experiments with the human genome, it was able to effectively discover novel pre-miRNAs, even in a future time lapse.

Funding

This work was supported by ANPCyT (PICT 2018 #3384 and PICT 2018 #2905) and UNL (CAI+D 2016 #082 and CAID 2020 #115). We also acknowledged the support of NVIDIA Corporation for the donation of Titan V GPU used for this research.

References

Amin, N. *et al.* (2019). Evaluation of deep learning in non-coding RNA classification. *Nature Machine Intelligence*, **1**(5), 246–256.

Bahdanau, D. *et al.* (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bartel, D. P. (2004). MicroRNAs. *Cell*, **116**(2), 281–297.

Bengio, Y. *et al.* (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(8), 1798–1828.

Bugnon, L. *et al.* (2019). Genome-wide hairpins datasets of animals and plants for novel mirna prediction. *Data in Brief*, **25**, 104209.

Bugnon, L. A. *et al.* (2020). Genome-wide discovery of pre-miRNAs: comparison of recent approaches based on machine learning. *Briefings in Bioinformatics*.

Chaabane, M. *et al.* (2019). circDeep: deep learning approach for circular RNA classification from other long non-coding RNA. *Bioinformatics*, **36**(1), 73–80.

de O. N. Lopes, I. *et al.* (2014). The discriminant power of RNA features for pre-miRNA recognition. *BMC Bioinformatics*, pages 124–134.

Devlin, J. *et al.* (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. cite arxiv:1810.04805Comment: 13 pages.

Dosovitskiy, A. *et al.* (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

Eraslan, G. *et al.* (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, **20**(7), 389–403.

He, K. *et al.* (2016). Identity mappings in deep residual networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham. Springer International Publishing.

Hofacker, I. L. (2003). Vienna rna secondary structure server. *Nucleic acids research*, **31**(13), 3429–3431.

Jurtz, V. I. *et al.* (2017). An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, **33**(22), 3685–3690.

LeCun, Y. *et al.* (2015). Deep learning. *Nature*, **521**(7553), 436–444.

Park, S. *et al.* (2017). Deep recurrent neural network-based identification of precursor micrornas. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2895–2904.

Raad, J. *et al.* (2020). Complexity measures of the mature miRNA for improving pre-miRNAs prediction. *Bioinformatics*, **36**(8), 2319–2327.

Saito, T. *et al.* (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, **10**(3).

Senior, A. W. *et al.* (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, **577**(7792), 706–710.

Seo, S. *et al.* (2018). DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, **34**(13), i254–i262.

Stegmayer, G. *et al.* (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*, **20**(5), 1607–1620.

Tang, X. and Sun, Y. (2019). Fast and accurate microRNA search using CNN. *BMC Bioinformatics*, **20**(S23).

Trieu, H.-L. *et al.* (2020). DeepEventMine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, **36**(19), 4910–4917.

Tsubaki, M. *et al.* (2018). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**(2), 309–318.

Vaswani, A. *et al.* (2017). Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Yones, C. *et al.* (2015). miRNAfe: A comprehensive tool for feature extraction in microRNA prediction. *Biosystems*, **138**, 1–5.

Yones, C. *et al.* (2020). HextractoR: an r package for automatic extraction of hairpins from genome-wide data. *bioRxiv*.

Yones, C. *et al.* (2021). High precision in microrna prediction: A novel genome-wide approach with convolutional deep residual networks. *Computers in Biology and Medicine*, **134**, 104448.

Zeng, H. *et al.* (2016). Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, **32**(12), i121–i127.

Zheng, X. *et al.* (2019). Nucleotide-level convolutional neural networks for pre-miRNA classification. *Scientific Reports*, **9**(1).

miRe2e: a full end-to-end deep model based on Transformers for prediction of pre-miRNAs from raw genome-wide data

J. Raad, L. Bugnon, D.H. Milone and G. Stegmayer

Research Institute for Signals, Systems and Computational Intelligence sinc(i) (FICH-UNL/CONICET), Ciudad Universitaria, Santa Fe, Argentina.

Contact: jraad@sinc.unl.edu.ar

Supplementary Material

PyTorch[1] was used to build and train the deep learning models. Our models were trained on a Nvidia Titan V GPU with 12 Gb of RAM. The architecture of the neural models are detailed in the following tables. We evaluated several loss functions, optimizers and learning rates on training data. The selected loss functions were: Mean Square Error (MSE) for the Structure prediction model and the MFE estimation model; and Focal loss (Lin *et al.*, 2017) with $\alpha = 1.0$ and $\gamma = 4.0$ for the pre-miRNA classifier. The optimizer selected was Stochastic Gradient Descent (SGD) with Nesterov momentum (Sutskever *et al.*, 2013), and a learning rate of 10^{-3}. More details about the architecture and learning configuration can be obtained from the source code[2].

[1] https://pytorch.org/

[2] https://github.com/sinc-lab/miRe2e

Table 1: Structure predictor.

Layer (type)	Output shape	Param #
ReLU-1	[4, 100]	0
BatchNorm1d-2	[4, 100]	8
Conv1d-3	[111, 100]	1,443
ReLU-4	[111, 100]	0
BatchNorm1d-5	[111, 100]	222
Conv1d-6	[111, 100]	37,074
ReLU-7	[111, 100]	0
BatchNorm1d-8	[111, 100]	222
Conv1d-9	[111, 100]	37,074
ResNet-10	[111, 100]	0
ReLU-11	[111, 100]	0
BatchNorm1d-12	[111, 100]	222
Conv1d-13	[111, 100]	37,074
ReLU-14	[111, 100]	0
BatchNorm1d-15	[111, 100]	222
Conv1d-16	[111, 100]	37,074
ResNet-17	[111, 100]	0
ReLU-18	[111, 100]	0
BatchNorm1d-19	[111, 100]	222
Conv1d-20	[111, 100]	37,074
ReLU-21	[111, 100]	0
BatchNorm1d-22	[-1, 111, 100]	222
Conv1d-23	[111, 100]	37,074
ResNet-24	[111, 100]	0
EncoderStr-25	[111, 100]	0
MultiheadAttention-26	[[2, 222], [100, 100]]	0
Dropout-27	[2, 222]	0
LayerNorm-28	[2, 222]	444
Linear-29	[2, 888]	198,024
Dropout-30	[2, 888]	0
Linear-31	[2, 222]	197,358
Dropout-32	[2, 222]	0
LayerNorm-33	[2, 222]	444
TransformerEncoderLayer-34	[2, 222]	0
MultiheadAttention-35	[[2, 222], [100, 100]]	0
Dropout-36	[2, 222]	0
LayerNorm-37	[2, 222]	444
Linear-38	[2, 888]	198,024
Dropout-39	[2, 888]	0
Linear-40	[2, 222]	197,358
Dropout-41	[2, 222]	0
LayerNorm-42	[2, 222]	444
TransformerEncoderLayer-43	[2, 222]	0
MultiheadAttention-44	[[2, 222], [100, 100]]	0
Dropout-45	[2, 222]	0
LayerNorm-46	[2, 222]	444
Linear-47	[2, 888]	198,024
Dropout-48	[2, 888]	0
Linear-49	[2, 222]	197,358
Dropout-50	[2, 222]	0
LayerNorm-51	[2, 222]	444
TransformerEncoderLayer-52	[2, 222]	0
MultiheadAttention-53	[[2, 222], [100, 100]]	0
Dropout-54	[2, 222]	0
LayerNorm-55	[2, 222]	444
Linear-56	[2, 888]	198,024
Dropout-57	[2, 888]	0
Linear-58	[2, 222]	197,358
Dropout-59	[2, 222]	0
LayerNorm-60	[2, 222]	444
TransformerEncoderLayer-61	[2, 222]	0
MultiheadAttention-62	[[2, 222], [100, 100]]	0
Dropout-63	[2, 222]	0
LayerNorm-64	[2, 222]	444
Linear-65	[2, 888]	198,024
Dropout-66	[2, 888]	0
Linear-67	[2, 222]	197,358
Dropout-68	[2, 222]	0
LayerNorm-69	[2, 222]	444
TransformerEncoderLayer-70	[2, 222]	0
MultiheadAttention-71	[[2, 222], [100, 100]]	0
Dropout-72	[2, 222]	0
LayerNorm-73	[2, 222]	444
Linear-74	[2, 888]	198,024
Dropout-75	[2, 888]	0
Linear-76	[2, 222]	197,358
Dropout-77	[2, 222]	0
LayerNorm-78	[2, 222]	444

TransformerEncoderLayer-79	[2, 222]	0
TransformerEncoder-80	[2, 222]	0
Dropout-81	[100, 222]	0
Linear-82	[100, 100]	22,300
ELU-83	[100, 100]	0
Dropout-84	[100, 100]	0
Linear-85	[100, 10]	1,010
ELU-86	[100, 10]	0
Linear-87	[100, 1]	11
Tanh-88	[100, 1]	0

Table 2: MFE estimation model.

Layer (type)	Output shape	Param #
ReLU-1	[5, 100]	0
BatchNorm1d-2	[5, 100]	10
Conv1d-3	[64, 100]	1,024
ReLU-4	[64, 100]	0
BatchNorm1d-5	[64, 100]	128
Conv1d-6	[64, 100]	12,352
ReLU-7	[64, 100]	0
BatchNorm1d-8	[64, 100]	128
Conv1d-9	[64, 100]	12,352
ResNet-10	[64, 100]	0
AvgPool1d-11	[64, 50]	0
ReLU-12	[64, 50]	0
BatchNorm1d-13	[64, 50]	128
Conv1d-14	[64, 50]	12,352
ReLU-15	[64, 50]	0
BatchNorm1d-16	[64, 50]	128
Conv1d-17	[64, 50]	12,352
ResNet-18	[64, 50]	0
AvgPool1d-19	[64, 25]	0
ReLU-20	[64, 25]	0
BatchNorm1d-21	[64, 25]	128
Conv1d-22	[64, 25]	12,352
ReLU-23	[64, 25]	0
BatchNorm1d-24	[64, 25]	128
Conv1d-25	[64, 25]	12,352
ResNet-26	[64, 25]	0
AvgPool1d-27	[64, 12]	0
Encoder-28	[64, 12]	0
Linear-29	[100]	76,900
ELU-30	[100]	0
BatchNorm1d-31	[100]	200
Linear-32	[30]	3,030
ELU-33	[30]	0
BatchNorm1d-34	[30]	60
Linear-35	[1]	31
ELU-36	[1]	0

Table 3: Pre-miRNA classifier.

Layer (type)	**Output shape**	**Param #**
ReLU-1	[5, 100]	0
BatchNorm1d-2	[5, 100]	10
Conv1d-3	[64, 100]	1,024
ReLU-4	[64, 100]	0
BatchNorm1d-5	[64, 100]	128
Conv1d-6	[64, 100]	12,352
ReLU-7	[64, 100]	0
BatchNorm1d-8	[64, 100]	128
Conv1d-9	[64, 100]	12,352
ResNet-10	[64, 100]	0
ReLU-11	[64, 100]	0
BatchNorm1d-12	[64, 100]	128
Conv1d-13	[64, 100]	12,352
ReLU-14	[64, 100]	0
BatchNorm1d-15	[64, 100]	128
Conv1d-16	[64, 100]	12,352
ResNet-17	[64, 100]	0

ReLU-18	[64, 100]	0
BatchNorm1d-19	[64, 100]	128
Conv1d-20	[64, 100]	12,352
ReLU-21	[64, 100]	0
BatchNorm1d-22	[64, 100]	128
Conv1d-23	[64, 100]	12,352
ResNet-24	[64, 100]	0
AvgPool1d-25	[64, 50]	0
ReLU-26	[64, 50]	0
BatchNorm1d-27	[64, 50]	128
Conv1d-28	[64, 50]	12,352
ReLU-29	[64, 50]	0
BatchNorm1d-30	[64, 50]	128
Conv1d-31	[64, 50]	12,352
ResNet-32	[64, 50]	0
ReLU-33	[64, 50]	0
BatchNorm1d-34	[64, 50]	128
Conv1d-35	[64, 50]	12,352
ReLU-36	[64, 50]	0
BatchNorm1d-37	[64, 50]	128
Conv1d-38	[64, 50]	12,352
ResNet-39	[64, 50]	0
ReLU-40	[64, 50]	0
BatchNorm1d-41	[64, 50]	128
Conv1d-42	[64, 50]	12,352
ReLU-43	[64, 50]	0
BatchNorm1d-44	[64, 50]	128
Conv1d-45	[64, 50]	12,352
ResNet-46	[64, 50]	0
AvgPool1d-47	[64, 25]	0
ReLU-48	[64, 25]	0
BatchNorm1d-49	[64, 25]	128
Conv1d-50	[64, 25]	12,352
ReLU-51	[64, 25]	0
BatchNorm1d-52	[64, 25]	128
Conv1d-53	[64, 25]	12,352
ResNet-54	[64, 25]	0
ReLU-55	[64, 25]	0
BatchNorm1d-56	[64, 25]	128
Conv1d-57	[64, 25]	12,352
ReLU-58	[64, 25]	0
BatchNorm1d-59	[64, 25]	128
Conv1d-60	[64, 25]	12,352
ResNet-61	[64, 25]	0
ReLU-62	[64, 25]	0
BatchNorm1d-63	[64, 25]	128
Conv1d-64	[64, 25]	12,352
ReLU-65	[64, 25]	0
BatchNorm1d-66	[64, 25]	128
Conv1d-67	[64, 25]	12,352
ResNet-68	[64, 25]	0
AvgPool1d-69	[64, 12]	0
ReLU-70	[64, 12]	0
BatchNorm1d-71	[64, 12]	128
Conv1d-72	[64, 12]	12,352
ReLU-73	[64, 12]	0
BatchNorm1d-74	[64, 12]	128
Conv1d-75	[64, 12]	12,352
ResNet-76	[64, 12]	0
ReLU-77	[64, 12]	0
BatchNorm1d-78	[64, 12]	128
Conv1d-79	[64, 12]	12,352
ReLU-80	[64, 12]	0
BatchNorm1d-81	[64, 12]	128
Conv1d-82	[64, 12]	12,352
ResNet-83	[64, 12]	0
ReLU-84	[64, 12]	0
BatchNorm1d-85	[64, 12]	128
Conv1d-86	[64, 12]	12,352
ReLU-87	[64, 12]	0
BatchNorm1d-88	[64, 12]	128
Conv1d-89	[64, 12]	12,352
ResNet-90	[64, 12]	0
AvgPool1d-91	[64, 6]	0
Encoder-92	[64, 6]	0
PositionalEncoder-93	[6, 64]	0
MultiheadAttention-94	[[2, 64], [6, 6]]	0
Dropout-95	[2, 64]	0
LayerNorm-96	[2, 64]	128

Linear-97	[2, 256]	16,64
Dropout-98	[2, 256]	0
Linear-99	[2, 64]	16,448
Dropout-100	[2, 64]	0
LayerNorm-101	[2, 64]	128
TransformerEncoderLayer-102	[2, 64]	0
MultiheadAttention-103	[[2, 64], [6, 6]]	0
Dropout-104	[2, 64]	0
LayerNorm-105	[2, 64]	128
Linear-106	[2, 256]	16,64
Dropout-107	[2, 256]	0
Linear-108	[2, 64]	16,448
Dropout-109	[2, 64]	0
LayerNorm-110	[2, 64]	128
TransformerEncoderLayer-111	[2, 64]	0
MultiheadAttention-112	[[2, 64], [6, 6]]	0
Dropout-113	[2, 64]	0
LayerNorm-114	[2, 64]	128
Linear-115	[2, 256]	16,64
Dropout-116	[2, 256]	0
Linear-117	[2, 64]	16,448
Dropout-118	[2, 64]	0
LayerNorm-119	[2, 64]	128
TransformerEncoderLayer-120	[2, 64]	0
TransformerEncoder-121	[2, 64]	0
BatchNorm1d-122	[385]	770
Linear-123	[1000]	386
ELU-124	[1000]	0
BatchNorm1d-125	[1000]	2
Dropout-126	[1000]	0
Linear-127	[1000]	1,001,000
Linear-128	[1000]	1,001,000
Linear-129	[2]	2,002
Softmax-130	[2]	0

References

Lin, T.-Y. *et al.* (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814.

Sutskever, I. *et al.* (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1139–1147.

Doctorado en Ingeniería
Mención en Inteligencia Computacional, Señales y Sistemas

Título de la obra:

Nuevos enfoques basados en medidas de complejidad para la detección de secuencias cortas en bioinformática

Autor: Jonathan Raad

Lugar: Santa Fe, Argentina

Palabras Claves:

medidas de complejidad, aprendizaje maquinal, aprendizaje profundo, predicción de microRNA, genoma completo.