Desarrollo de un metaclasificador de dianas de microRNA

Antonio Herbello Rodríguez
Máster en Bioinformática y Bioestadística
TFM-Bioinformática y Bioestadística Área 3

Consultor: Albert Pla Planas
PARA: Ferran Prados Carrasco

24/06/2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Descripción del trabajo*
Nombre del autor:	*Antonio Herbello Rodríguez*
Nombre del consultor/a:	*Albert Pla Planas*
Nombre del PRA:	*Ferran Prados Carrasco*
Fecha de entrega (mm/aaaa):	06/2020
Titulación::	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*TFM-Bioinformática y Bioestadística Área 3*
Idioma del trabajo:	*Español*
Palabras clave	*microRNA, metaclassifier, bioinformatics*

Resumen del Trabajo (máximo 250 palabras):

Este trabajo se ha desarrollado con el objetivo de crear meta modelo capaz de predecir dianas de microRNAs en humanos, mejorando la actuación de los predictores utilizados.

Desde el descubrimiento de los microRNAs hace casi 30 años se ha ido viendo el importante papel que estas secuencias de ~22 nucleótidos tienen en todos los procesos celulares y en el desarrollo de numerosas enfermedades. Para determinar el impacto de un miRNA en nuestro organismo es fundamental conocer la diana sobre la que actúa.

Con este objetivo se han desarrollado numerosos métodos computacionales pero estos, en general, carecen de la exactitud necesaria. El meta learning ha demostrado mejorar las predicciones en los problemas de machine learning a los que se le ha enfrentado.

Teniendo esto en cuenta se ha hecho una selección de predictores de dianas y se han entrenado varios modelos de machine learning con los resultados de los diferentes clasificadores. Se ha obtenido un meta modelo que mejora las predicciones de todos los predictores empleados.

Abstract (in English, 250 words or less):

The main object of this project was to create a meta model capable of predicting microRNA targets in humans. Improving the performance of the predictors involved.

Since the discovery of microRNAs almost 30 years ago, the important role that these ~22 nucleotide sequences have in all cellular processes and in the development of numerous diseases has been found. To determine the impact of a miRNA on our body, it is essential to know the target on which the miRNA acts.

Many computational metods has been developed with this objective. Generally this methods give too much noise. Meta Learning has shown how it is able to outperform base learners on machine learning tasks.

With this in consideration, a miRNA predictor selection has been done and multiples machine learning models has been trained with the results of the predictors. Finally, a meta model has been obtained. This meta model outperforms the base learners used for its training.

Índice

Listado de tablas

Listado de ilustraciones

1. Resumen

1.1 Contexto y justificación del Trabajo

Desde el descubrimiento en 1993[1] de los miRNA, su actividad se ha relacionado con los mecanismos moleculares de varias enfermedades y con la regulación de potencialmente toda la actividad celular[2].

Para determinar la actividad de un miRNA es fundamental el conocimiento de los sitios de unión miRNA-mRNA. La realización de experimentos in vivo para determinar estos sitios de unión está muy limitada. Esta situación ha potenciado el desarrollo de herramientas bioinformáticas (muchas de estas usan algoritmos de machine learning) capaces de predecir las dianas sobre las que actúan los miRNA, aunque en general presentan mucho ruido[3].

Diferentes meta modelos han demostrado ser capaces de mejorar los resultados de las predicciones realizadas por clasificadores individuales. Por lo tanto, un meta clasificador de dianas basado en ensemble learning podría mejorar las predicciones de los predictores individuales que se utilicen[4].

1.2 Objetivos del Trabajo

El objetivo principal es el desarrollo de un metaclasificador de dianas de microRNA.

Los sub objetivos que se contemplan son:

-Evaluar métodos de predicción existentes.
-Definición de un dataset de entrenamiento con suficientes casos negativos y positivos.
-Evaluar los métodos con los que se puede implementar un metaclasificador.

1.3 Enfoque y método seguido

En primer lugar se identificaron diferentes predictores de dianas ya existentes, estos tenían que cumplir ciertos requisitos como complementariedad en el tipo de datos que utilizan, que fuesen métodos descargables y ejecutables en el ordenador personal, la posibilidad de reentrenamiento. Finalmente estos criterios fueron desechados por el reducido número de clasificadores que se pudo instalar.

El dataset de entrenamiento y validación ha sido adaptado de uno ya existente debido a las dificultades que se estaban presentando durante su desarrollo. Este dataset se ha empleado para evaluar los predictores elegidos y los diferentes meta modelos, estos últimos mediante la realización de la técnica K fold cross-validation.

Por último se ha implementado el meta clasificador elegido (k-Nearest Neighbors) en R y se ha creado un script que realiza todo el proceso, paso de los datos por los primeros clasificadores, recolección de los resultados y presentación de estos al meta clasificador.

1.4 Planificación del Trabajo

La planificación inicial del trabajo que se recoge en la PEC1 ha sido considerablemente retrasada. Esto se achaca a dificultades técnicas en la mayoría de los apartados y asuntos personales acrecentados por la crisis del COVID-19.

La planificación final ha quedado:

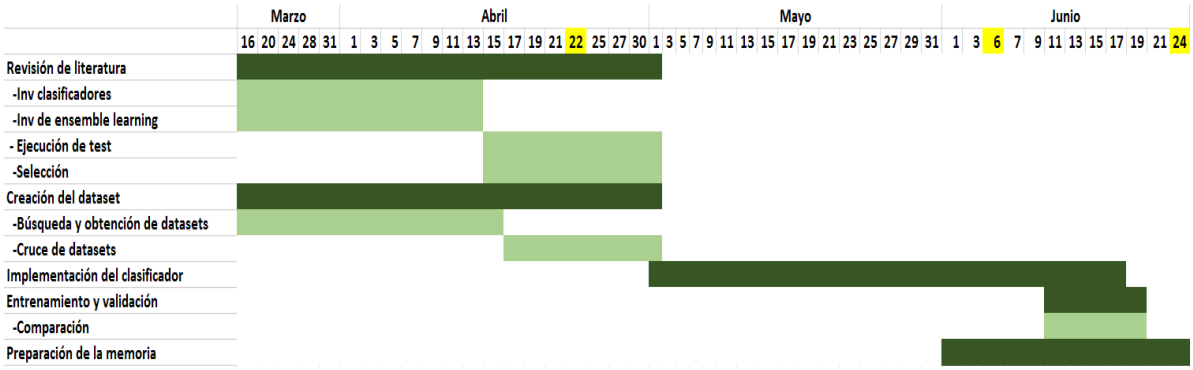

Ilustración 1. Planificación final

1.5 Breve sumario de productos obtenidos

-Script de R (clasificador.R): script con el que se ejecutan los diferentes predictores de dianas y el meta clasificador entrenado.

-Modelo entrenado (modelo_knn1.rds): archivo .rds que contiene el modelo entrenado y se puede cargar para hacer predicciones con él.

-Presentación virtual

-Memoria

1.6 Breve descripción de los otros capítulos de la memoria

- Introducción: breve explicación teórica de diferentes conceptos relacionados con el trabajo.

- Metodología: diferentes pasos que se han seguido en la realización del trabajo.

- Resultados: análisis de los diferentes modelos evaluados

- Conclusión: consideración de los resultados obtenidos en relación a los objetivos fijados y posibles ampliaciones del trabajo.

2. Introducción

2.1 MicroRNA

2.1.1 ¿Qué es un micro RNA?

Un miRNA es una pequeña molécula endógena de ARN monocatenario con una longitud aproximada de 22 nucleótidos, que se encarga de regular la expresión génica a nivel post transcripcional.

Los genes miRNA constituyen una de las familias de genes más abundantes, presente en animales, plantas, protistas y virus[5]. En humanos, la mayoría se encuentran codificados en intrones, codificantes o no, aunque también pueden aparecer en exones.[6]

2.1.2 Biogénesis

Normalmente los genes miRNA aparecen próximos entre ellos formando una unidad de transcripción policistrónica. Los miRNA pertenecientes a la misma unidad son co-transcritos, pero pueden ser individualmente regulados a nivel post-transcripcional[7].

Los genes miRNA son transcritos por la RNA polimerasa II (RNA pol III se ha observado en algunos virus) este proceso está regulado por los factores de transcripción asociados a RNA pol II y reguladores epigenéticos. Se genera un transcrito primario (pri miRNA) con un tamaño que puede variar de unos cientos de bases a decenas de miles[6].

Este transcrito, que aún permanece en el núcleo, es reconocido por un complejo de varias proteínas conocido como Microprocessor. Sus componentes principales son la RNAsa III Drosha y el dominio de unión a RNA DGGCR8/Pasha. Su función es cortar el transcrito primario para generar una horquilla más corta (alrededor de 70 nucleótidos) que constituye el pre-miRNA. Este intermediario sale del núcleo al citoplasma, donde una RNAsa III, Dicer, lo corta definiendo el otro extremo del miRNA y formando el dúplex miR/miR*. Finalmente la cadena menos estable en 5' del dúplex se une al complejo de silenciamiento inducido por RNA (RISC), mientras que la miR* se degrada[8]. Este proceso se puede apreciar en la ilustración 2.

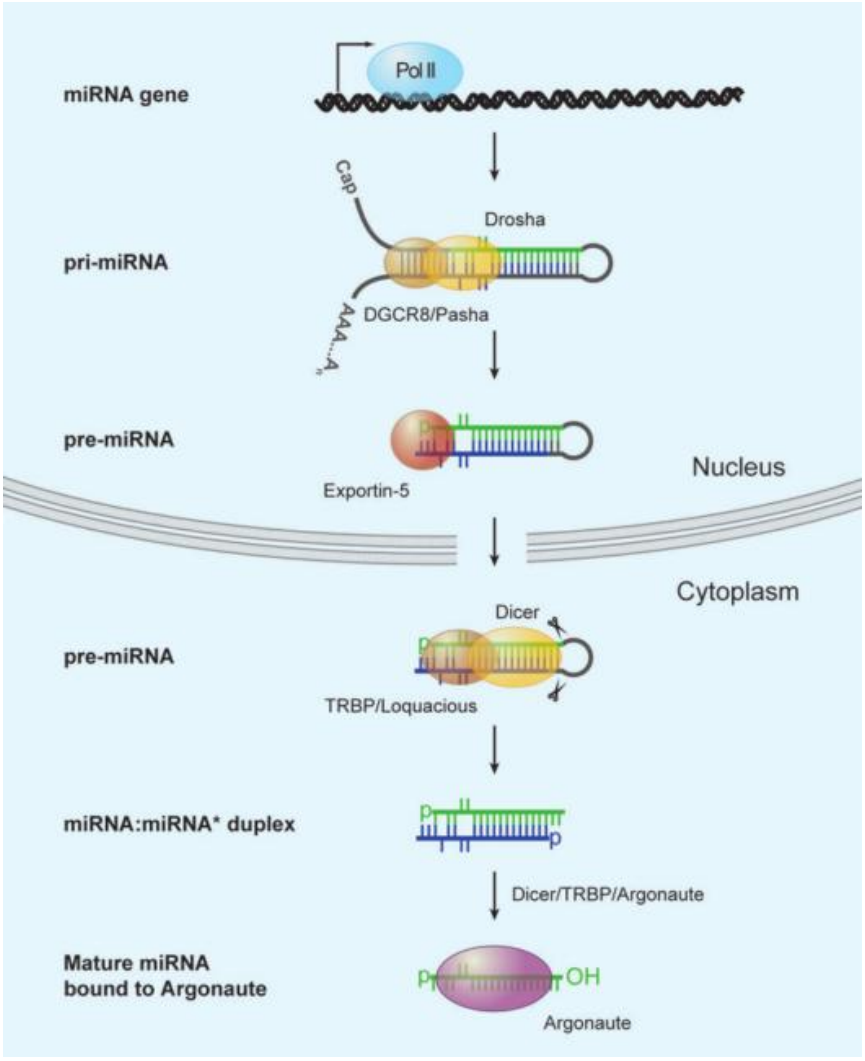

Ilustración 2. Biogénesis de los microRNA. Imagen tomada del estudio realizado por Natascha Busati y Stephen M. Cohen (2007)[8]

2.1.3 Funciones

Los miRNA han sido relacionados con la regulación de la mayoría de funciones celulares(diferenciación, desarrollo, metabolismo, proliferación, apoptosis, infección viral) además se ha observado como la expresión alterada de estos está involucrada en el desarrollo de numerosas enfermedades, entre ellas el cáncer [2].

Cuando un miRNA actúa sobre un mRNA codificante de una proteína, su expresión se ve reducida, ya sea por la degradación del mRNA o por la inhibición de la traducción. Apreciándose un descenso en los niveles de esa proteína en lugar de un silenciamiento absoluto. La reducción de la expresión se puede intensificar con la unión de varios miRNAs o uniéndose a los mRNA codificantes de varias proteínas de una misma ruta.[9]

Los miRNA pueden reprimir la expresión de sus dianas de forma directa. Para esto existen varios mecanismos: complementariedad de bases en la seed region o en la region central.

2.2 Machine Learning

2.2.1 ¿Qué es el machine learning?

El machine learning (ML) consiste en una serie de algoritmos computacionales diseñados para emular la inteligencia humana aprendiendo del entorno. Con la gran cantidad de datos que existen actualmente, el machine learning ha sido fundamental en todos los campos, desde las finanzas hasta la biología. El uso que los algoritmos hace de estos datos permite su conversión en conocimiento, algo de otra forma inimaginable[10].

2.2.2 Tipos de algoritmos machine learning

Existen tres tipos clases principales de algoritmos en machine learning:

-Aprendizaje supervisado (supervised learning): En el aprendizaje supervisado los datos que se suministran están etiquetados, de estos datos el modelo debe aprender para poder hacer predicciones en datos futuros[11].

Supervised learning usa técnicas de clasificación y regresión para elaborar los modelos predictivos. Entre los algoritmos comunes destacan: kNN, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machine (SVM).

-Aprendizaje no supervisado (unsupervised learning): En el aprendizaje no supervisado no existe ningún tipo de etiqueta en los datos suministrados, de forma que el modelo debe aprender por si mismo las relaciones existentes entre los datos para poder hacer predicciones[11].

La técnica más común es clustering. Se usa para encontrar motivos escondidos en los datos. Los algoritmos más comunes son: K-means clustering, hierachical clustering, hidden Markov models

-Reinforcement learning: En el aprendizaje reforzado se busca desarrollar un sistema que mejore su actuación a partir de las interacciones con el entorno. Es similar al supervised learning pero los datos suministrados no contienen la etiqueta.

2.2.3 Algoritmos más usados

A continuación se explica brevemente algunos de los algoritmos más usados y que han sido testeados en este trabajo:

-Logistic Regression: es un algoritmo similar al de regresión lineal que se utiliza para problemas de clasificación. Al contrario que en un modelo lineal la variable

respuesta puede ser tanto categórica como continua. Existen tres categorías: binomial, multinomial y ordinal[12].

-Árbol de decisión: es uno de los algoritmos de supervised learning más importantes. Son árboles en los que cada nodo representa una característica para ser clasificada y cada rama representa el valor que ese nodo toma. La clasificación se inicia en el nodo de la raíz y va subiendo por el árbol. Sus ventajas principales son: produce resultados intensivos, fácil de entender y tiene una estructura de conocimiento bien organizada[13]. Con los arboles de decisión también se construyen modelos Random Forest.

-kNN: k-nearest neighbor es uno de los algoritmos de supervised learning más usados para clasificaciones en las que hay poco o ningún conocimiento previo sobre la distribución de los datos. La clasificación final depende de la mayoría de las categorías asignadas.

-SVM: Support Vector Machines comprenden un grupo de métodos que son usados tanto para clasificación como para regresión. Sus beneficios son: 1) efectivos en espacios de altas dimensiones, 2) eficientes en cuanto a la memoria que usan, 3) es muy versátil, diferentes funciones kernel pueden ser utilizadas[13].

-Naive Bayes: algoritmo muy usado para labores de clasificación. La principal ventaja de este tipo de métodos es que requieren poco training data para estimar los parámetros necesarios para la clasificación.

-Random Forest: método de ensemble learning muy extendido, en el que se usan varios árboles de decisión para reducir la varianza. El método por el que se combinan los árboles de decisión es el baggin, de forma que cada árbol se entrena con datos diferentes. Son usados tanto en regresión como clasificación.

-Neural Network: conjunto de algoritmos que intentan emular el cerebro humano, diseñados para reconocer patrones en los datos. Pueden usarse tanto en clustering como clasificación[12].

2.2.4 Meta Learning

El Meta Learning es un subcampo del ML que se encarga del estudio de cómo se puede incrementar la eficiencia en los sistemas de ML a través de la experiencia, de forma que el aprendizaje sea flexible al entorno[14,15]. Un meta learner aprende del conocimiento aprendido por otros learners (base learners) y se adapta con la experiencia[16].

Métodos

Algunos de los métodos más utilizados basados en ensemble learning:

-Stacked generalisation (stacking): este método consiste en la utilización de los resultados de diferentes base learners sobre un mismo dataset como input para

un learner. De esta forma se construye un meta modelo que relacionará los resultados de los base learners con el valor real[16]. Por regla general los pasos a seguir son: (1) Separar el dataset en dos. (2) Entrenar varios base learners con uno de los sub datasets. (3) Testear los base learners con el otro subset. (4) Entrenar el meta learners con los resultados del testeo de los base learners como input[17]. Al separar el dataset en dos se está evitando el over-fitting (aprender muy bien de los datos de entrenamiento pero presentar problemas al generalizar) del meta clasificador, esto se daría si se usase el mismo dataset para entrenar los base learners y el meta learner.

-Boosting: el objetivo del boosting es crear un clasificador fuerte a partir de varios clasificadores débiles. Para este método el conjunto de base learners débiles seleccionado es entrenado con diferentes variantes del mismo dataset cambiado los pesos[18]. Los base learners se van entrenado sucesivamente, de forma que el base learner n+1 se centra en los datos que el base learner n no ha sido capaz de clasificar[16].

-Bagging: primero se crean múltiples datasets del dataset original, de forma que cada sub dataset es un subset aleatorio del original. Después se entrena un algoritmo de machine learning con cada dataset. La predicción final es la combinación, normalmente por votación o las medias, de los resultados de los diferentes algoritmos[16,19].

2.3 Predictores de dianas

Desde hace décadas se vienen desarrollando multitud de métodos para la predicción de dianas de microRNA. La identificación de dianas sobre las que actúa un miRNAs es esencial para poder determinar la función de este, tanto si su expresión es normal como si es aberrante[20].

Actualmente miles de dianas han sido identificadas, pero faltan muchísimas más. Aproximaciones in vivo como PCRs, hibridación in situ o Northern blot se han intentado para resolver este problema, pero los resultados no han sido satisfactorios. En general, los experimentos en laboratorio para validar posibles dianas de miRNA son muy costosos y poco eficientes, esto ha propiciado el desarrollo de métodos computacionales[21–23].

2.3.1 Características

Hay cuatro características principales que utilizan la mayoría de los métodos para predecir dianas: (1)Seed Match. Unión entre el miRNA y la seed region del mRNA, es por complementariedad Watson-Crick (A-U,G-C). (2)Conservación. Se da cuando una secuencia es común entre diferentes especies. Se busca tanto en el extremo 3’ UTR (untraslated region) como en el 5’ UTR. (3)Energía libre. Si la unión miRNA:mRNA es estable termodinámicamente, se considera más probable que se trate de una diana. (4)Accesibilidad. Medida de la facilidad con la que se pueda dar la unión miRNA-mRNA. La estructura secundaria que toma

el mRNA tras la transcripción puede alterar la facilidad del miRNA para unirse[20,24].

La primera generación de herramientas de predicción se basaba en las anteriores características, las tres primeras mayormente. Posteriormente con la realización de estudios de expresión CLIP-seq, la detección de SNPs en las regiones diana o los clúster de familias miRNA ha ido aumentando el conocimiento sobre la interacción miRNA-mRNA. Esto ha propiciado la aparición de herramientas basadas en machine learning, que han sido capaces de englobar un mayor número de características aumentando así la precisión de las predicciones[20,24].

La mayor dificultad que han encontrado las herramientas basadas en machine learning son los datos negativos para su entrenamiento. Si los datos negativos contienen secuencias muy artificiales el modelo probablemente no será entrenado correctamente y si son demasiado parecidas a los miRNA reales es posible que no sea capaz de distinguir positivos de negativos[25]. Aunque se han desarrollado clasificadores que utilizan sólo datos positivos, el uso de ambos tipos de datos mejora la actuación del clasificador[26].

3. Metodología

3.1 Herramientas

El software elegido para realizar el trabajo ha sido RStudio[27]. RStudio es un IDE para el lenguaje R[28] que está disponible de forma gratuita. La versión utilizada de R ha sido la 4.0.0. Se ha elegido esta herramienta por la familiaridad previa (ha sido la herramienta principal a lo largo del máster), por la facilidad con la que se pueden observar los datos gracias a su interfaz gráfica y por las posibilidades que ofrece rmarkdown a la hora de secuenciar y visualizar el trabajo.

Los paquetes que se han empleado son:

-caret[29]: se ha usado para hacer el cross-validation, crear los modelos y para obtener la mayoría de las métricas.
-multiMir[30]: se ha usado como predictor.
-seqinr[31]: se ha usado para escribir los archivos fasta.
-precrec[32]: se ha usado para crear las curvas ROC y obtener los valores de AUC.
-dplyr[33]: se ha usado a la hora de procesar los datos.

3.2 Predictores de dianas

3.2.1 Predictores seleccionados

Para la elección de clasificadores adecuados, se utilizó como guía una de las tablas comparativas del material suplementario del estudio de Kern, F. *et al.*(2019)[34]. En estas tablas se recogen un total de 98 herramientas para la predicción de dianas y muchas de sus características (si es descargable, el organismo diana, tipos de input y output, características de los mirs en que se basan…).

Se seleccionaron los métodos basados en machine learning, descargables en los que tanto organismo de la diana como del miRNA incluían el *Homo sapiens*. También se optó por el uso del paquete de R Multimir, debido al reducido número de clasificadores que quedó.

-IntaRNA[35]: es un programa implmentado en C++ y Perl, predice interacciones RNA-RNA sin restricción de la región. Requiere que se introduzcan 2 secuencias en formato fasta. Al introducir un archivo fasta con varias secuencias de miRNA y otro con las secuencias dianas el programa hace todos los cruzamientos posibles.

Para realizar sus predicciones se basa en la energía libre de hibridación de los RNA implicados y la energía libre necesaria para abrir los sitios de interacción.

En el caso de una predicción positiva el programa devuelve la energía y un dibujo de la interacción.

-TarPmiR[36]: es un programa implementado en Python, basado en random forest, que toma como inputs las secuencias del miRNA y la diana, además del corte de probabilidad elegido por el usuario. Se pueden introducir múltiples secuencias en formato fasta.

Para realizar las predicciones usa información de las secuencias, la accesibilidad del sitio de interacción, la estructura y la conservación. Como resultados devuelve un tabla con varias características de cada unión miRNA-mRNA algunas de esta son: la probabilidad de la unión, el sitio de unión, la energía, el contenido de AU…

-miRanda[37]: es un programa implementado en C que se basa en dynamic programming para hacer las predicciones. Al igual que en los otros dos predictores el input que acepta son la secuencia del microRNA y la del mRNA diana, pudiéndose introducir varias en formato fasta.

Los principios que utiliza para realizar las predicciones son la complementariedad de secuencias y la estructura. Como IntaRNA sus resultados incluyen una representación de la unión y la energía libre, además de otras características del enlace como la posición.

-Paquete MultiMir[30]: este paquete de R incluye información de varias databases relacionando miRNAs con sus dianas, enfermedades y medicamentos. Una de las opciones que presenta es restringir la búsqueda a una base de datos realizada con las predicciones de varios predictores de dianas. Los predictores que incluye son: Diana micro T, TargetScore, miRanda, ElMMo, PITA, PicTar y MicroCosm.

3.2.2 Predictores descartados

La lista inicial de clasificadores antes de poder ser reducida por criterios como el rendimiento o las características que utilizaban, se redujo considerablemente por dificultades a la hora de instalarlos y hacerlos funcionar. En tabla 1 se puede ver una lista de predictores de dianas descartados con la razón de descarte.

Descartados	
Clasificador	Razón para descarte
TargetSpy[38]	Problemas con viennaRNA
TargetThermo[39]	Problemas con viennaRNA
TargetExpress[40]	Necesita valores de expresión, no funciona la descarga de predicciones
microCLIP[41]	Problemas con viennaRNA
SVMicrO[42]	No encuentra ./svmicro.pl

chimiRic[43]	No hay manual, no funciona
Expmicro[44]	No hay manual, no funciona
Cupid[45]	Necesita matlab
GenMir++[46]	No va la web, no lo he podido conseguir
IMTRBM[47]	No hay manual
miRepress[48]	No va la web, no lo he podido conseguir
MIRZA[49]	Devuelve el output vacío
SeedVicious[50]	Falla con los datos test que trae

Tabla 1. Predictores descartados y la razón de descarte.

A esta lista se le suman tres clasificadores más: TargetMiner[51], MBStar[52] y MultiMiTar[53]. Estos tres clasificadores fueron instalados correctamente, cuando se testearon con los datos test que incluyen se obtenían resultados, pero en cuanto se probaron con datos diferentes empezaron los errores. Las dos modalidades de error que comparten los tres eran: "Run successfully" con el output en blanco o nunca acababan de correr.

3.3 Datos

Los datos usados forman parte de los datos suplementarios que presenta miRAW[54]. Este dataset se centra en miRNAs de humanos, para su realización se usaron las bases de datos Diana Tar Base[55] y MirTarBase[56] , además de la construcción de un dataset de ejemplos negativos[54]. El dataset original se completa con datos de las localizaciones de las interacciones miRNA-mRNA obtenidos por experimentos PAR-Clip[57] y CLASH[58] y con la entradas de Ensembl[59], pero para los clasificadores utilizados en este trabajo no eran necesarios.

En concreto se ha utilizado el archivo "allTrainingsites.txt", este contiene el nombre de miRNA, gen, Ensembl id, secuencia del miRNA, secuencia del transcrito diana y la validación de 65427 muestras (33143 positivas y 32284 negativas).

3.3.1 Procesado de los datos

En primer lugar se eliminaron todas las entradas que contenían algún NA, tenían algún valor vacío o contenían en las secuencias alguna L, ya que los predictores de dianas utilizados dejaban de funcionar cuando se encontraban con alguno de estos casos. 65427 muestras iniciales pasaron a 62215(30140 negativos y 32075 positivos).

El volumen del dataset tuvo que ser reducido debido a la dificultad que tenían los predictores elegidos para procesar todos los datos. Se tomó alrededor del 15% del dataset original, 9450 muestras (ver anexo 3). Este fue dividido de manera aleatoria en 10 sub datasets para facilitar el procesado de los datos en los clasificadores.

Tres de los clasificadores originalmente seleccionados requerían el Refseq id de los transcritos. Con el paquete de R biomart[60] se realizó una búsqueda de estos con el Ensembl id correspondiente a cada gen. Finalmente estos cambios fueron descartados una vez se comprobó que no funcionaban los clasificadores que los requerían.

Antes de pasar los datos por los clasificadores se prepararon para estos. Se les añadió un columna con un id único para poder seguir el resultado de las diferentes comprobaciones que realizan y se cambió de columnas algunos datos que tenían el gen del nombre cambiado por el Ensembl id. Después se crearon dos archivos fasta por sub dataset, uno con el nombre del miRNA y su secuencia y el otro con el id y la secuencia del transcrito de RNA. Para la creación de los archivos fasta se utilizó el paquete Seqinr de R.

Para la búsqueda en el paquete multiMir se aislaron los Ensembl id en un vector. El cual se pasó por la función de búsqueda (get_multimir()), como output de esta búsqueda se obtiene un data frame con todas las interacciones de las diferentes bases de datos asociadas a los predictores.

3.3.2 Procesado de los resultados

Los resultados al salir de cada clasificador de dianas tenían un formato diferente, esto supuso que en los tres casos se tuvieran que desarrollar códigos en R para adaptarlos.

Para los resultados de TarPmiR se utilizó el id único de cada entrada del dataset para identificar los cruzamientos miRNA-mRNA y se anotó la probabilidad (entre 0 y 1) que el programa atribuía a cada interacción.

Con miRanda e IntaRNA se siguió un proceso similar al de TarPmiR, en estos programas se obtiene energía libre de las interacciones positivas, por lo que sólo se anotó 1 (interacción) o 0 (no interacción).

Los resultados obtenidos con el paquete multiMir se dejaron como los devuelve el programa. Dependiendo del predictor se obtiene una probabilidad entre 0 y 1 o la energía libre.

Por cada subset (10 en total) del dataset inicial se obtuvieron 9 datasets de resultados, uno por clasificador, considerando los diferentes clasificadores del paquete MultiMir independientes. Estos 90 datasets fueron unificados en uno que contuviese una columna atribuida a los resultados de cada clasificador.

3.4 Entrenamiento de modelos

Se han entrenado varios posibles modelos como meta clasificador. Todos los modelos usados pertenecen al paquete de R caret. En el anexo 1 se muestra el código necesario para realizarlo.

Antes de empezar el entrenamiento de modelos se utilizó una de las funciones (trainControl()) del paquete caret para realizar K fold cross-validation. La k elegida fue 10, este método implica la subdivisión del dataset en k (10) sub datasets con los que se entrena y testea el modelo.

A continuación se especifica el modelo probado, los parámetros afinados y entre comillas el nombre por el que se llama al método en la función de caret:

-k-Nearest Neighbors ("kknn"): para este modelo el único parámetro que se ha afinado es la Kmáx, se han probado los valores 5, 7 y 9.

-Optimal Weighted Nearest Neighbor Classifier ("ownn"): en este caso se ha afinado la K con los valores 9 y 5.

-L2 Regularized Linear Support Vector Machines with Class Weights ("svmLinearWeights2"): tres variables se han afinado en este modelo cost(0.25,0.5,1), weights (1,2,3) y loss(L1,L2).

-Least Squares Support Vector Machine with Radial Basis Function Kernel ("lssvmRadial"): los parámetros afinados han sido sigma y tau. Para sigma los valores han sido: (1)0.0262103084548232, (2)0.134841392378085 y (3) 0.243472476301347. Para tau: (1) 0.0625, (2)0.125 y (3) 0.25. El número que cada valor tiene delante sirve para identificarlos en la tabla de resultados.

-Conditional Inference Random Forest ("cforest"): el único parámetro afinado es mtry con valores de 2,5 y 9.

-Neural Network ("nnet"): dos parámetros se han afinado para este modelo, decay (0, 1e-04, 0.1) y size (1, 3, 5).

-Boosted Logistic Regression ("LogitBoost"): se han probado 11, 21 y 31 iteraciones.

-Bayesian Generalized Linear Model ("bayesglm"): no se ha afinado ningún parámetro para este modelo ya que no presenta ninguno para afinar.

4. Resultados

4.1 Predictores de dianas

En este apartado se va a mostrar el rendimiento obtenido en los predictores de diana seleccionados con el dataset original. Para comprobarlo se ha reestructurado el dataset, las energías y las probabilidades se han sustituido por 1, considerándose interacción positiva.

Las métricas que se han considerado para este apartado son 3 (accuracy, sensitivity y specificity), el valor de estas va a servir para compararlas con el meta modelo.

Accuracy (exactitud): predicciones correctas respecto al total de las muestras probadas.
Sensitivity (sensibilidad): verdaderos positivos etiquetados como positivos.
Specificity (especificidad): verdaderos negativos etiquetados como negativos.
Precision (precisión): media de sensitivity y specificity
F-Score (valor F): medida que considera precision y recall.

Herramienta	Accuracy	Sensitivity	Specificity
Inta RNA	0.7687	0.6964	0.8348
TarPmiR	0.5844	0.7768	0.4274
miRanda	0.8015	0.99	0.6289
MM-TargetScore	0.6012	0.9816	0.2529
MM-Dianamicrot	0.7151	0.9167	0.5306
MM-PITA	0.5796	0.9128	0.2746
MM-PICTAR	0.5247	0.99756	0.09181
MM-microcosm	0.4915	0.99247	0.03304
MM-Elmmo	0.7498	0.8837	0.6273

Tabla 2. Métricas de los predictores de dianas utilizados.

4.2 Evaluación de modelos

A continuación se presentan una serie de tablas y gráficos. Cada una de estas tablas se corresponde con un modelo entrenado y contiene las métricas para

diferentes ajustes de los parámetros. El mejor ajuste está marcado en verde. Los gráficos son curvas ROC(AUC) y Precision-Recall del modelo con los ajustes marcados en cada caso.

Las métricas que se han tenido en cuenta son Accuracy, Specificity, Sensitivity, F-score y Precision para comparar los modelos con diferente afinado. Despues se ha introducido AUC(ROC) para comparar el mejor de cada modelo, a estos también se les han realizado las curvas ROC y Precision-Recall (izquierda y derecha respectivamente en las ilustraciones).

Todas las métricas menos AUC(ROC) se han obtenido con las función confusionMatrix() del paquete caret. AUC(ROC) y las gráficas se han obtenido con las funciones evalmod() y auc() del paquete precrec.

- k-Nearest Neighbors (knn1)

Kmax	Accuracy	Specificity	Sensitivity	F-Score	Precision
9	0.8781	0.8022	0.9610	0.88281	0.81640
7	0.8693	0.8184	0.9249	0.87120	0.82337
5	0.847	0.8161	0.8806	0.84617	0.81429

Tabla 3. Métricas para knn1.

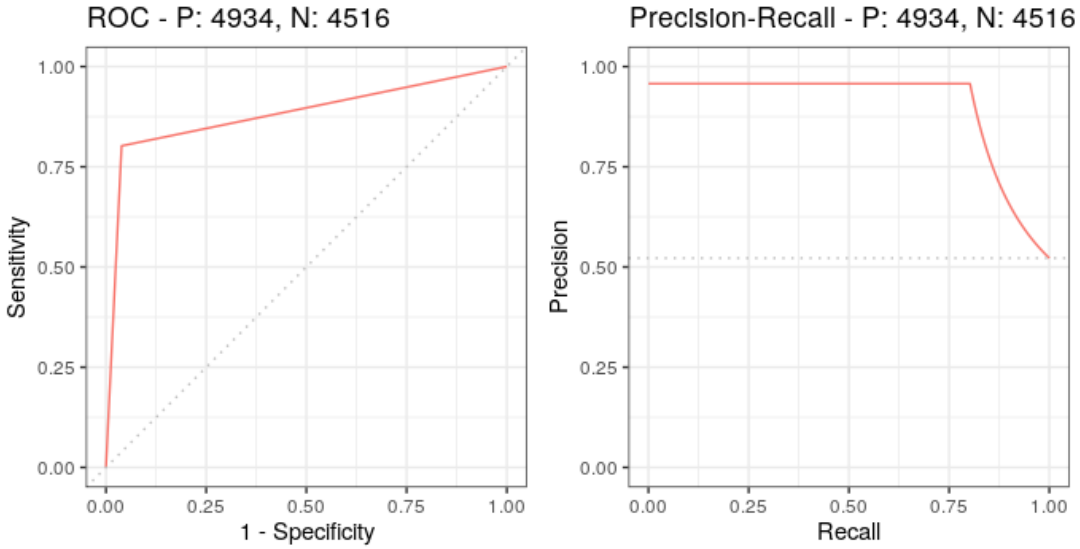

Ilustración 3. Curvas ROC y Precision-Recall del modelo knn1 marcado..

- Optimal Weighted Nearest Neighbor Classifier (knn2)

K	Accuracy	Specificity	Sensitivity	F-Score	Precision
9	0.8748	0.7932	0.9639	0.88037	0.81016
5	0.8779	0.7991	0.9639	0.88296	0.81455

Tabla 4. Métricas para knn2.

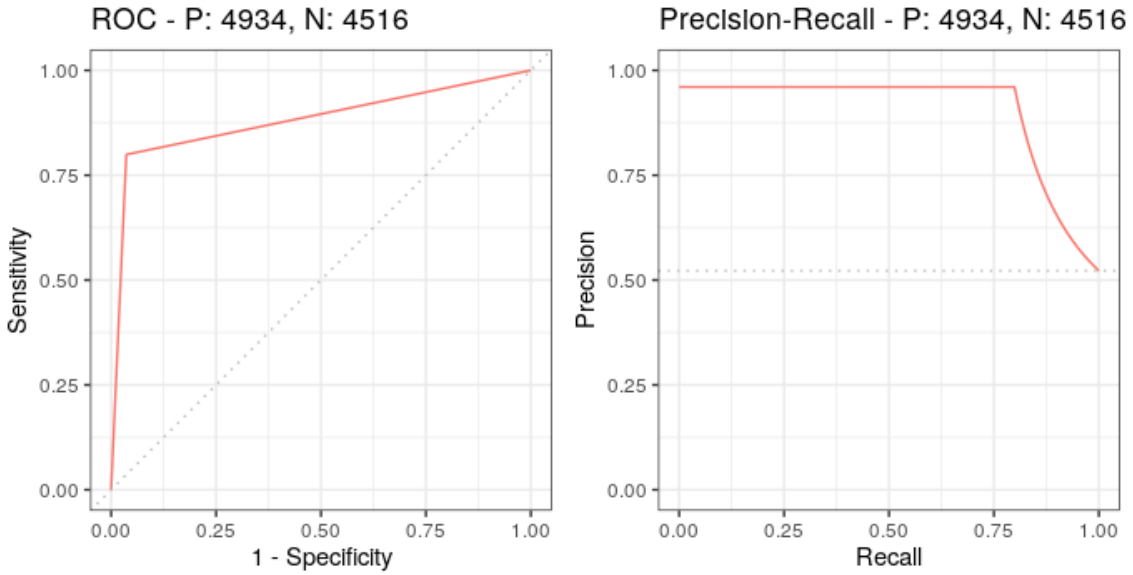

Ilustración 4. Curvas ROC y Precision-Recall para el modelo knn2 marcado.

- L2 Regularized Linear Support Vector Machines with Class Weights (SVM1)

C	W	L	Accuracy	Specificity	Sensitivity	F-Score	Precision
0.25	1	L1	0.838	0.7604	0.9227	0.8448	0.79902
0.5	1	L1	0.8375	0.7610	0.9209	0.8441	0.79913
1	1	L1	0.8384	0.7610	0.9229	0.8451	0.77950
0.25	2	L1	0.8326	0.8575	0.8053	0.8213	0.83801
0.5	2	L1	0.8317	0.8573	0.8038	0.8203	0.83756
1	2	L1	0.8286	0.8548	0.7998	0.8168	0.83456
0.25	3	L1	0.7863	0.8942	0.6685	0.7494	0.85258
0.5	3	L1	0.786	0.8942	0.6678	0.7489	0.85245
1	3	L1	0.7863	0.8935	0.6691	0.7495	0.85198
0.25	1	L2	0.8494	0.7673	0.9391	0.8563	0.78697
0.5	1	L2	0.8493	0.7673	0.9388	0.8562	0.78693
1	1	L2	0.8493	0.7673	0.9388	0.8562	0.78693
0.25	2	L2	0.839	0.8431	0.8345	0.8321	0.82962
0.5	2	L2	0.8392	0.8433	0.8345	0.8321	0.82981
1	2	L2	0.8393	0.8433	0.8348	0.8323	0.82984
0.25	3	L2	0.795	0.8889	0.6924	0.7635	0.85088
0.5	3	L2	0.7951	0.8891	0.6924	0.7636	0.85111
1	3	L2	0.7952	0.8891	0.6926	0.7637	0.85115

Tabla 5. Métricas para los diferentes parámetros de SVM1.

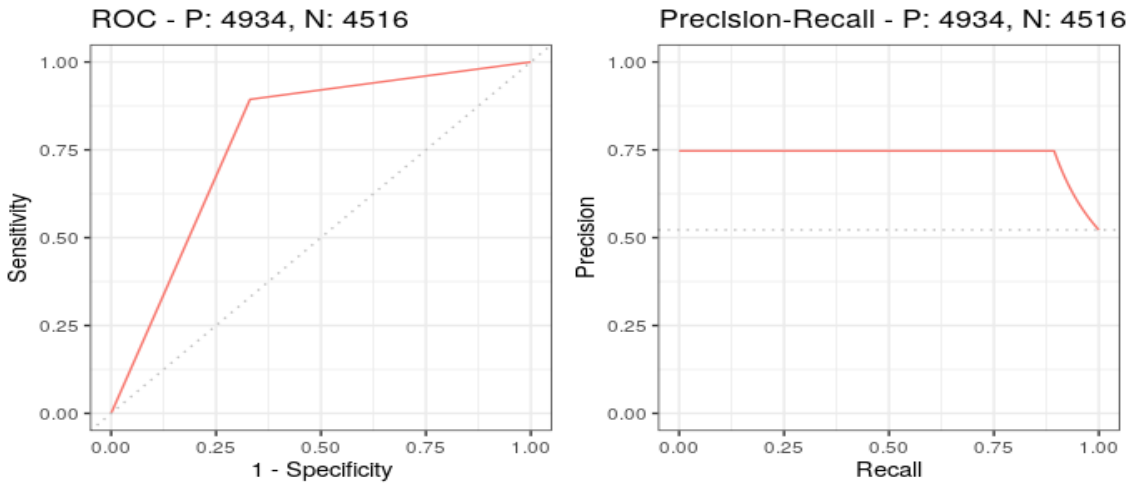

Ilustración 5. Curvas ROC y Precision-Recall del modelo SVM1 marcado.

- Least Squares Support Vector Machine with Radial Basis Function Kernel (SVM2)

Sigma	Tau	Accuracy	Specificity	Sensitivity	F-Score	Precision
1	1	0.7415	0.9098	0.5575	0.6733	0.84981
1	2	0.7415	0.9100	0.5573	0.6732	0.85005
1	3	0.7412	0.9104	0.5562	0.6725	0.85037
2	1	0.7061	0.9217	0.4705	0.6048	0.84627
2	2	0.7061	0.9217	0.4705	0.6048	0.84627
2	3	0.7063	0.9221	0.4705	0.6049	0.84695
3	1	0.6294	0.9422	0.2876	0.4259	0.82007
3	2	0.6294	0.9422	0.2876	0.4259	0.82007
3	3	0.6288	0.9426	0.2858	0.4239	0.82020

Tabla 6. Métricas para los diferentes parámetros del modelo SVM2.

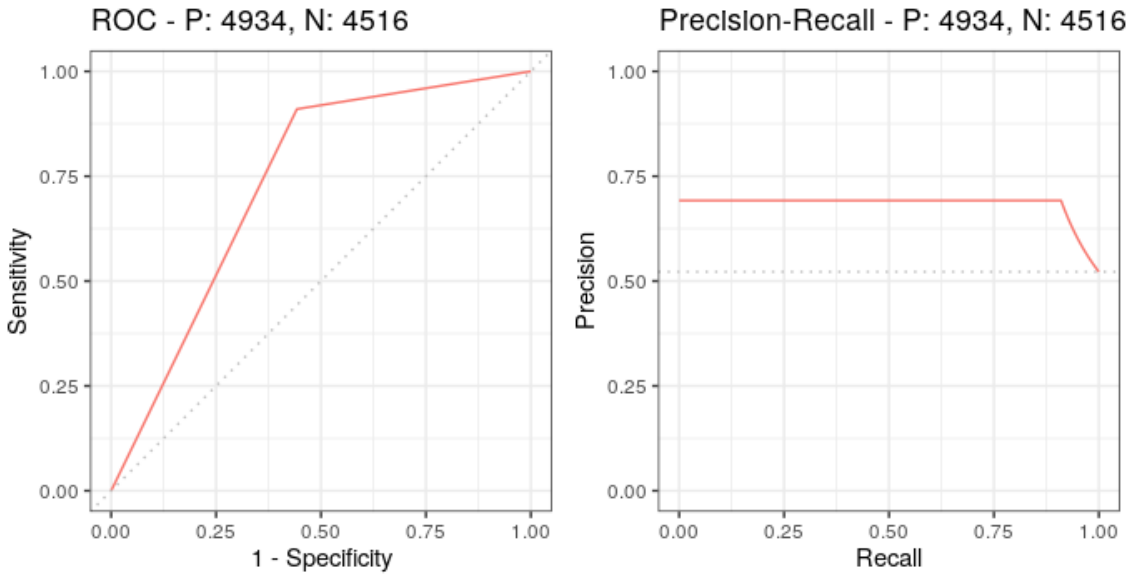

Ilustración 6. Curvas ROC y Precision-Recall del modelo SVM2 marcado.

- Conditional Inference Random Forest (RF)

Mtry	Accuracy	Specificity	Sensitivity	F-Score	Precision
2	0.8691	0.7902	0.9552	0.8746	0.80650
5	0.8759	0.8094	0.9484	0.8795	0.82002
9	0.8759	0.8117	0.9459	0.8792	0.82138

Tabla 7. Métricas para los diferentes parámetros del modelo RF.

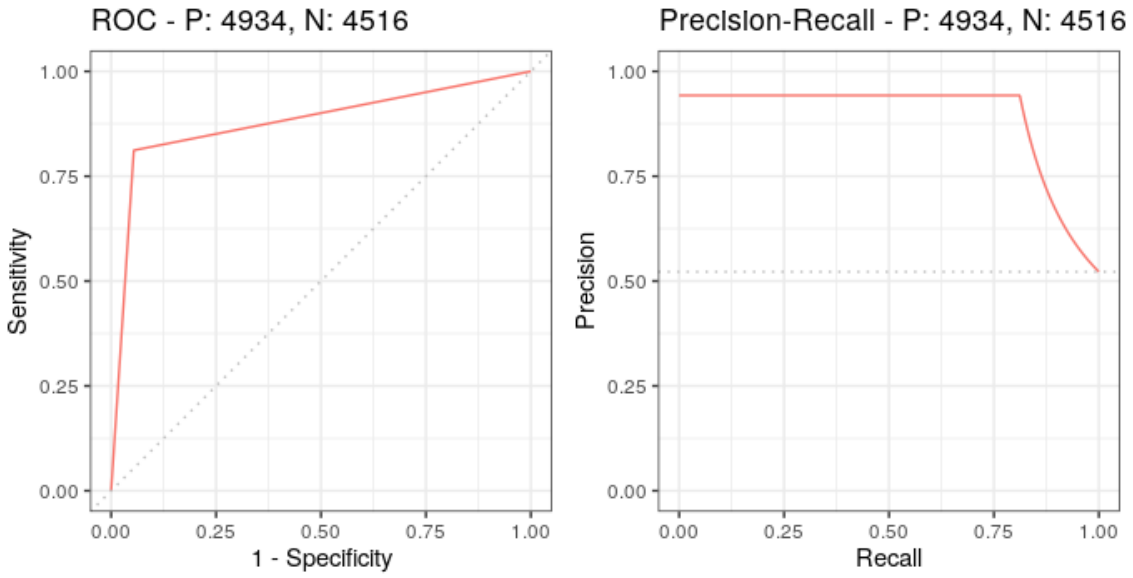

Ilustración 7. Curvas ROC y Precision-Recall del modelo RF marcado.

- Neural Network (NN)

Size	Decay	Accuracy	Specificity	Sensitivity	F-Score	Precision
1	0	0.8392	0.8032	0.8784	0.8392	0.80336
3	0	0.8624	0.7967	0.9342	0.8665	0.80792
5	0	0.865	0.8968	0.9284	0.8679	0.81480
1	1e-04	0.8381	0.7906	0.8899	0.8400	0.79552
3	1e-04	0.8599	0.7983	0.9271	0.8634	0.80798
5	1e-04	0.866	0.8032	0.9346	0.8695	0.81298
1	0.1	0.8599	0.7928	0.9331	0.8642	0.80481
3	0.1	0.8607	0.7995	0.9275	0.8642	0.80899
5	0.1	0.866	0.8046	0.9331	0.8694	0.81382

Tabla 8. Métricas para los diferentes parámetros del modelo NN.

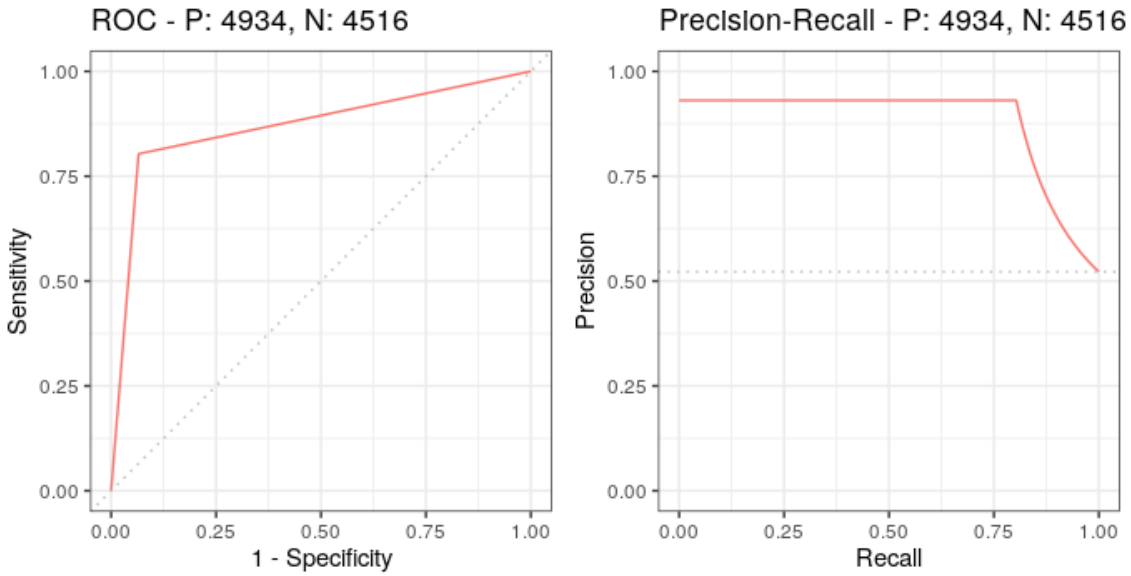

Ilustración 8. Curvas ROC y Precision-Recall del modelo NN marcado.

-Boosted logistic regression (BLR)

niter	Accuracy	Specificity	Sensitivity	F-Score	Precision
11	0.8601	0.7969	0.9291	0.8639	0.80723
21	0.8624	0.8042	0.9260	0.8654	0.81235
31	0.8605	0.8021	0.9242	0.8636	0.81048

Tabla 9. Métricas para los diferentes parámetros del modelo BLR.

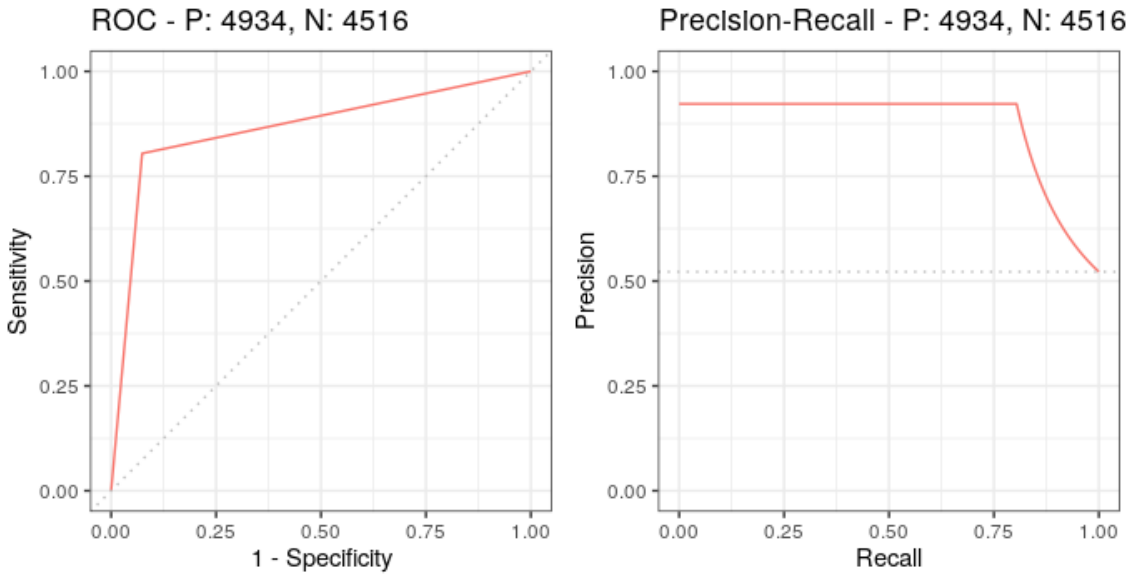

Ilustración 9. Curvas ROC y Precision-Recall del modelo BLR marcado.

-Bayesian Generalized Linear model (GLM)

	Accuracy	Specificity	Sensitivity	F-Score	Precision
GLM	0.8558	0.7902	0.9273	0.8600	0.80183

Tabla 10. Métricas para el modelo GLM.

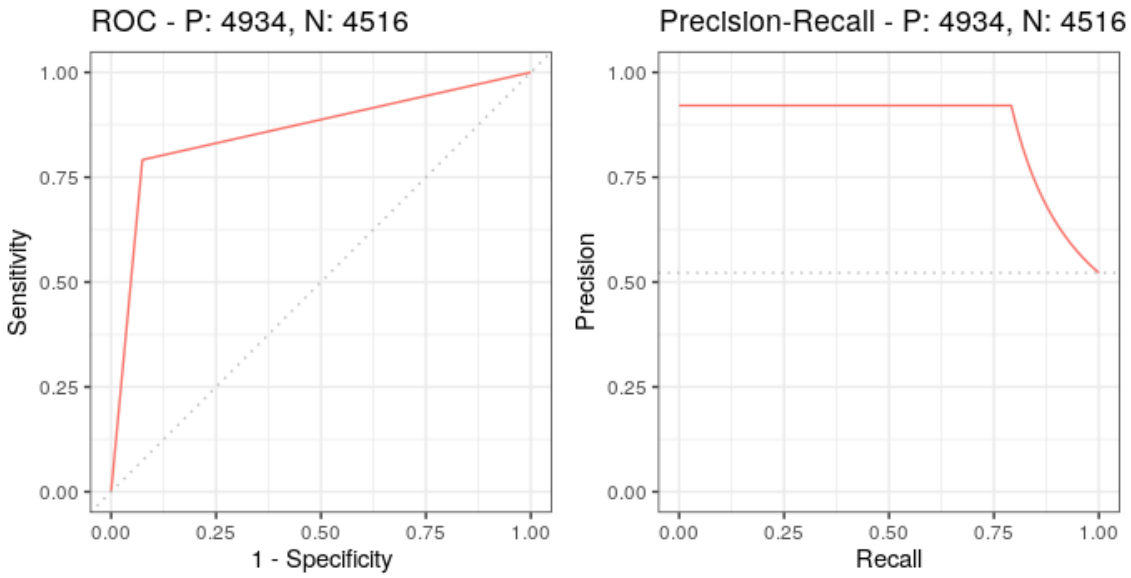

Ilustración 10. Curvas ROC y Precision-Recall del modelo GLM.

Una vez expuestos los resultados de los diferentes modelos con diferentes ajustes, se van a aislar los mejores modelos en una última tabla, en la que se marcará el modelo final elegido:

Modelo	Accuracy	Specificity	Sensitivity	F-Score	Precision	AUC(ROC)
knn1	0.8781	0.8022	0.9610	0.88281	0.81640	0.8816
knn2	0.8779	0.7991	0.9639	0.88296	0.81455	0.8815
SVM1	0.8494	0.7673	0.9391	0.8563	0.78697	0.7813
SVM2	0.7415	0.9100	0.5573	0.6732	0.85005	0.7336
RF	0.8759	0.8117	0.9459	0.8792	0.82138	0.8788
NN	0.866	0.8032	0.9346	0.8695	0.81298	0.8689
BLR	0.8624	0.8042	0.9260	0.8654	0.81235	0.8651
GLM	0.8558	0.7902	0.9273	0.8600	0.80183	0.8683

Tabla 11. Métricas de los mejores modelos de cada tipo.

Con estos resultados se decidió elegir el modelo marcado knn1, que se corresponde con el modelo k-Nearest Neighbors con kmax = 9.

En la tabla 11 se puede ver como tiene la mayor accuracy y AUC(ROC) de todos los contemplados. En el resto de métricas rondan los valores más altos observados. En general, todos los modelos probados han dado buenos resultados, SVM2 ha sido el único modelo en obtener valores por debajo de los observados en los predictores de dianas.

4.3 Implementación

Una vez se eligió el modelo con mejores resultados este se guardó en un archivo .rda con las función saveRDA(). Además se desarrolló un script de R que realiza el proceso completo necesario para la utilización del modelo. El script se llama con dos argumentos, los dos archivos en formato csv. El primero tiene que contener el nombre del miRNA y su secuencia, y el segundo el Ensembl id del mRNA u otro identificador y su secuencia.

Este script se encarga de ir llamando a los diferentes clasificadores involucrados, crear los archivos necesarios para su funcionamiento, puesta en común de los resultados, adaptación de los resultados para el modelo, realización de las predicciones y por último, devuelve un archivo en formato csv con las predicciones (predicciones.txt).

Para su funcionamiento es necesario:

-Tener instalados los predictores de diana TarPmiR, miRanda e IntaRNA.
-Poseer el archivo que contiene el modelo (modelo_knn1.rds).
-Conexión a internet para poder hacer uso del paquete multiMir.
-Modificar las variables que contienen las direcciones de cada clasificador y del modelo.

El script solo se ha probado en Ubuntu 18.04.4 LTS. Está pensando para ejecutarse desde el terminal con la llamada:

```
Rscript --vanilla clasificador.R testmir.txt testmrna.txt
```

En el anexo 2 juntos con el script y el modelo se adjuntan dos archivos de test con sólo 10 secuencias miRNA y mRNA.

5. Conclusiones

En cuanto al cumplimiento de los objetivos marcados se concluye:

1. La realización del meta clasificador de dianas de miRNA se ha completado con éxito.

Se ha conseguido un meta clasificador que alimentándolo sólo con los resultados de predictores de dianas es capaz de mejorar las predicciones de todos estos, obteniendo una AUC (ROC) de 0.8816, se puede decir que se ha obtenido un buen clasificador. En general todos los modelos probados han mostrado buenos resultados, mientras que los dos modelos knn han mostrados los mejores resultados, los SVM han sido los peores. En especial SVM2, que no ha superado a algunos de los predictores de dianas empleados (Inta RNA, miRanda).

2. La definición del dataset no se ha completado.

Las dificultades que conllevaba la realización del dataset estaba retrasando mucho el trabajo, por lo que se optó por utilizar uno ya existente. A pesar de esto, algunas de las tareas incluidas en este objetivo se han completado, como el cruce de diferentes datasets. Esta tarea se realizó con éxito para obtener los Refseq ID de los transcritos, pero finalmente dejaron de ser necesarios y se desecharon.

3. La evaluación de los métodos existentes se ha completado.

La selección de los predictores ha sido pobre. Hubiese sido mucho más interesante combinar predictores más variados, lo más actuales posibles y con posibilidad de ser reentrenados. Las dificultades que fueron surgiendo con la instalación y puesta en marcha de los predictores estaba retrasando mucho el trabajo. El número de predictores instalados y probados con éxito era muy bajo por lo que se decidió utilizar el paquete multiMir. Esto estaba contemplado como medida de contención en caso de que algo similar ocurriese poder seguir avanzando en la realización del trabajo.

4. La evaluación de los métodos para la implantación del meta clasificador se ha completado.

De los diferentes métodos considerados inicialmente se decidió utilizar Stacking, esta técnica de ensemble learning combina las predicciones de diferentes clasificadores para entrenar un meta clasificador. Teniendo en cuenta el uso de un solo dataset, la diferente naturaleza de los clasificadores y los resultados obtenidos se considera como la elección acertada.

En cuanto a los productos obtenidos se concluye:

-El script final a pesar de cumplir su función tiene mucho margen para mejora. El tiempo para su realización ha estado muy limitado debido al retraso que llevaba

el trabajo. Ahora mismo está muy restringido en cuanto al tipo de input que acepta (sólo .csv con dos columnas, la primera con el nombre y la segunda con la secuencia), además el código para procesar los resultados puede ser realmente lento cuando los archivos analizados son extensos.

Finalmente, una vez con que el trabajo ha finalizado, plantearía la repetición de la selección de predictores con un mayor número de predictores. Esta nueva selección podría cubrir un mayor número de características y más variedad en el tipo de algoritmo. Se podrían incluir algunos de los predictores descartados por razones técnicas (ej. Expmicro, ComiR, Cupid, TargetExpress) o de los incluidos en el paquete multiMir (ej. Diana-microT-cds, TargetScore). De esta forma se podrían mejorar los resultados obtenidos.

6. Glosario

miRNA: micro ARN

mRNA: ARN mensajero

ROC: Receiver operating characteristic.

AUC: Area under the curve.

Knn(1,2): hace referencia a uno de los dos modelos K-nearest neighbors empleados.

GLM: hace referencia al modelo de regresión logística.

SVM(1,2): hace referencia a uno de los dos modelos Support Vector Machine empleados.

RF: hace referencia al modelo random forest empleado.

NN: hace referencia al modelo Neural Network empleado.

BLR: hace referencia al modelo bayesiano empleado.

7. Bibliografia

1. Lee, R. C., Feinbaum, R. L. & Ambros, V. The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell* **75,** 843–854 (1993).
2. Huang, Y. *et al.* Biological functions of microRNAs: A review. *Journal of Physiology and Biochemistry* **67,** 129–139 (2011).
3. Saçar, M. D. & Allmer, J. Machine learning methods for microRNA gene prediction. *Methods Mol. Biol.* **1107,** 177–187 (2014).
4. Yu, S., Kim, J., Min, H. & Yoon, S. Ensemble learning can significantly improve human microRNA target prediction. *Methods* **69,** 220–229 (2014).
5. Griffiths-Jones, S., Saini, H. K., Van Dongen, S. & Enright, A. J. miRBase: Tools for microRNA genomics. *Nucleic Acids Res.* **36,** 154–158 (2008).
6. Ha, M. & Kim, V. N. Regulation of microRNA biogenesis. *Nature Reviews Molecular Cell Biology* **15,** 509–524 (2014).
7. Lee, Y., Jeon, K., Lee, J. T., Kim, S. & Kim, V. N. MicroRNA maturation: Stepwise processing and subcellular localization. *EMBO J.* **21,** 4663–4670 (2002).
8. Bushati, N. & Cohen, S. M. microRNA Functions. *Annu. Rev. Cell Dev. Biol.* **23,** 175–205 (2007).
9. Mohr, A. M. & Mott, J. L. Overview of microRNA biology. *Seminars in Liver Disease* **35,** 3–11 (2015).
10. El Naqa, I. & Murphy, M. J. What Is Machine Learning? in *Machine Learning in Radiation Oncology* 3–11 (Springer International Publishing, 2015). doi:10.1007/978-3-319-18305-3_1
11. Machine Learning: Tres cosas que es necesario saber - MATLAB & Simulink. Available at: https://es.mathworks.com/discovery/machine-learning.html. (Accessed: 22nd June 2020)
12. Dreiseitl, S. & Ohno-Machado, L. Logistic regression and artificial neural network classification models: A methodology review. *J. Biomed. Inform.* **35,** 352–359 (2002).
13. Muhammad, I. & Yan, Z. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. doi:10.21917/ijsc.2015.0133
14. Vilalta, R. & Drissi, Y. *A Perspective View and Survey of Meta-Learning*. *Artificial Intelligence Review* **18,** (2002).
15. Vanschoren, J. Meta-Learning: A Survey. (2018).
16. Lemke, C., Budka, M. & Gabrys, B. Metalearning: a survey of trends and technologies. *Artif. Intell. Rev.* **44,** 117–130 (2015).
17. Usha Rani, D., Prasanna Kumari, G. & Professor, A. *A Study of Meta-Learning in Ensemble Based Classifier*. *An International Journal (ESTIJ)* **2,** (2012).
18. Vilalta, R., Giraud-Carrier, C. & Brazdil, P. Meta-Learning - Concepts and Techniques. in *Data Mining and Knowledge Discovery Handbook* 717–731 (Springer US, 2009). doi:10.1007/978-0-387-09823-4_36
19. Ensemble Methods (Part 1): Model averaging, Bagging and Random Forests | CommonLounge. Available at: https://www.commonlounge.com/discussion/9b90beb11aff4bf4ad628262ff27e06d. (Accessed: 22nd June 2020)
20. Peterson, S. M. *et al.* Common features of microRNA target prediction tools. *Front. Genet.* **5,** 1–10 (2014).

21. Grad, Y. *et al.* Computational and experimental identification of C. elegans microRNAs. *Mol. Cell* **11,** 1253–1263 (2003).
22. Chen, P. Y. *et al.* The developmental miRNA profiles of zebrafish as determined by small RNA cloning. *Genes Dev.* **19,** 1288–1293 (2005).
23. Kleftogiannis, D. *et al.* Where we stand, where we are moving: Surveying computational techniques for identifying miRNA genes and uncovering their regulatory role. *Journal of Biomedical Informatics* **46,** 563–573 (2013).
24. Watanabe, Y., Tomita, M. & Kanai, A. Computational Methods for MicroRNA Target Prediction. *Methods Enzymol.* **427,** 65–86 (2007).
25. Wu, Y., Wei, B., Liu, H., Li, T. & Rayner, S. MiRPara: A SVM-based software tool for prediction of most probable microRNA coding regions in genome scale sequences. *BMC Bioinformatics* **12,** 107 (2011).
26. Yousef, M., Jung, S., Showe, L. C. & Showe, M. K. Learning from positive examples when the negative class is undetermined- microRNA gene identification. *Algorithms Mol. Biol.* **3,** 2 (2008).
27. RStudio Team. RStudio: Integrated Development Environment for R. (2020).
28. R Core Team. R: A Language and Environment for Statistical Computing. (2020).
29. Kuhn, M. caret: Classification and Regression Training. (2020).
30. Ru, Y., Mulvahill, M., Mahaffey, S. & Kechris, K. multiMiR: Integration of multiple microRNA-target databases with their disease and drug associations.
31. Charif, D. & Lobry, J. R. Seqin{R} 1.0-2: a contributed package to the {R} project for statistical computing devoted to biological sequences retrieval and analysis. in *Structural approaches to sequence evolution: Molecules, networks, populations* (eds. Bastolla, U., Porto, M., Roman, H. E. & Vendruscolo, M.) 207–232 (Springer Verlag, 2007).
32. Saito, T. & Rehmsmeier, M. Precrec: fast and accurate precision-recall and ROC curve calculations in R. *Bioinformatics* **33 (1),** 145–147 (2017).
33. Wickham, H., François, R., Henry, L. & Müller, K. dplyr: A Grammar of Data Manipulation. (2020).
34. Kern, F. *et al.* What's the target: understanding two decades of in silico microRNA-target prediction. *Brief. Bioinform.* **0,** 1–12 (2019).
35. Busch, A., Richter, A. S. & Backofen, R. IntaRNA: Efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics* **24,** 2849–2856 (2008).
36. Ding, J., Li, X. & Hu, H. TarPmiR: A new approach for microRNA target site prediction. *Bioinformatics* **32,** 2768–2775 (2016).
37. Enright, A. J. *et al.* MicroRNA targets in Drosophila. *Genome Biol.* **5,** R1 (2003).
38. Sturm, M., Hackenberg, M., Langenberger, D. & Frishman, D. TargetSpy: A supervised machine learning approach for microRNA target prediction. *BMC Bioinformatics* **11,** 292 (2010).
39. Lekprasert, P., Mayhew, M. & Ohler, U. Assessing the Utility of Thermodynamic Features for microRNA Target Prediction under Relaxed Seed and No Conservation Requirements. *PLoS One* **6,** e20622 (2011).
40. Ovando-Vázquez, C., Lepe-Soltero, D. & Abreu-Goodger, C. Improving microRNA target prediction with gene expression profiles. *BMC Genomics* **17,** 364 (2016).

41. Paraskevopoulou, M. D., Karagkouni, D., Vlachos, I. S., Tastsoglou, S. & Hatzigeorgiou, A. G. microCLIP super learning framework uncovers functional transcriptome-wide miRNA interactions. *Nat. Commun.* **9,** 1–16 (2018).
42. Liu, H., Yue, D., Chen, Y., Gao, S. J. & Huang, Y. Improving performance of mammalian microRNA target prediction. *BMC Bioinformatics* **11,** 476 (2010).
43. Lu, Y. & Leslie, C. S. Learning to Predict miRNA-mRNA Interactions from AGO CLIP Sequencing and CLASH Data. *PLOS Comput. Biol.* **12,** e1005026 (2016).
44. Liu, H. *et al.* A Bayesian approach for identifying miRNA targets by combining sequence prediction and gene expression profiling. *BMC Genomics* **11,** S12 (2010).
45. Chiu, H. S. *et al.* Cupid: Simultaneous reconstruction of micrornatarget and cerna networks. *Genome Res.* **25,** 257–267 (2015).
46. Huang, J. C. *et al.* Using expression profiling data to identify human microRNA targets. *Nat. Methods* **4,** 1045–1049 (2007).
47. Liu, Y., Luo, J. & Ding, P. Inferring MicroRNA targets based on restricted boltzmann machines. *IEEE J. Biomed. Heal. Informatics* **23,** 427–436 (2019).
48. Ghosal, S. *et al.* MiRepress: Modelling gene expression regulation by microRNA with non-conventional binding sites. *Sci. Rep.* **6,** 1–13 (2016).
49. Khorshid, M., Hausser, J., Zavolan, M. & Van Nimwegen, E. A biophysical miRNA-mRNA interaction model infers canonical and noncanonical targets. *Nat. Methods* **10,** 253–255 (2013).
50. Marco, A. SeedVicious: Analysis of microRNA target and near-target sites. *PLoS One* **13,** e0195532 (2018).
51. Bandyopadhyay, S. & Mitra, R. TargetMiner: microRNA target prediction with systematic identification of tissue-specific negative examples. **25,** 2625–2631 (2009).
52. Bandyopadhyay, S., Ghosh, D., Mitra, R. & Zhao, Z. MBSTAR: Multiple instance learning for predicting specific functional binding sites in microRNA targets. *Sci. Rep.* **5,** 1–12 (2015).
53. Mitra, R. & Bandyopadhyay, S. MultiMiTar: A Novel Multi Objective Optimization based miRNA-Target Prediction Method. *PLoS One* **6,** e24583 (2011).
54. Pla, A., Zhong, X. & Rayner, S. miRAW: A deep learning-based approach to predict microRNA targets by analyzing whole microRNA transcripts. *PLOS Comput. Biol.* **14,** e1006185 (2018).
55. Vlachos, I. S. *et al.* DIANA-TarBase v7.0: Indexing more than half a million experimentally supported miRNA:mRNA interactions. *Nucleic Acids Res.* **43,** D153–D159 (2015).
56. Chou, C.-H. *et al.* miRTarBase 2016: updates to the experimentally validated miRNA-target interactions database. *Nucleic Acids Res.* **44,** 239–247 (2016).
57. Grosswendt, S. *et al.* Unambiguous Identification of miRNA: Target site interactions by different types of ligation reactions. *Mol. Cell* **54,** 1042–1054 (2014).
58. Helwak, A., Kudla, G., Dudnakova, T. & Tollervey, D. Mapping the human miRNA interactome by CLASH reveals frequent noncanonical binding. *Cell*

153, 654–665 (2013).
59. Aken, B. L., Ayling, S. & Barrell, D. The Ensembl gene annotation system. *Database* **2016,** (2016).
60. Durinck, S. *et al.* BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics* **21,** 3439–3440 (2005).

8. Anexo

Listado de archivos que se adjunta:

Anexo 1. Códigos R utilizados en el trabajo que no aparecen en el script final (archivo .Rmd).
Anexo 2. Archivo comprimido (.ZIP) que contiene el script final (archivo .R), el modelo entrenado (archivo .rds) y dos archivos test para el script (.txt).
Anexo 3. Dataset utilizado en formato csv (9450 entradas, 4516 negativas y 4934 positivas).