Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Maestría en Ciencias en Ciencias de la Computación

Clasificación multiclase de péptidos antimicrobianos: un enfoque comparativo

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de Maestro en Ciencias

Presenta:

Sergio Alejandro Pinacho Castellanos

Ensenada, Baja California, México

2021

Tesis defendida por

Sergio Alejandro Pinacho Castellanos

y aprobada por el siguiente Comité

Dr. Carlos Alberto Brizuela Rodríguez
Codirector de tesis

Dr. César Raúl García Jacas
Codirector de tesis

Dr. Israel Marck Martínez Pérez

Dr. Sergio Andrés Águila Puentes

Dr. Israel Marck Martínez Pérez
Coordinador del Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez
Directora de Estudios de Posgrado

Resumen de la tesis que presenta Sergio Alejandro Pinacho Castellanos como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Clasificación multiclase de péptidos antimicrobianos: un enfoque comparativo

Resumen aprobado por:

Dr. Carlos Alberto Brizuela Rodríguez	Dr. César Raúl García Jacas
Codirector de tesis	Codirector de tesis

La continua proliferación de agentes infecciosos capaces de resistir a los fármacos antimicrobianos, existentes en el mercado, se ha convertido en una de las mayores preocupaciones para la salud pública mundial. Los péptidos antimicrobianos (AMPs, por sus siglas en inglés) han ganado popularidad al considerarse candidatos prometedores para usarse solos o en combinación con los fármacos actuales para hacer frente a la resistencia antimicrobiana. Para asistir a la investigación y descubrimiento de nuevos AMPs con potencial uso terapéutico, se han propuesto modelos computacionales basados en algoritmos clásicos de aprendizaje de máquina. Un problema con estos modelos radica en que el costo computacional para la selección de un conjunto óptimo de descriptores moleculares, para representar los datos, crece exponencialmente con el número de descriptores disponibles inicialmente para la selección. Recientemente, modelos computacionales basados en redes neuronales de múltiples capas (conocidos como modelos profundos) mostraron que es posible eliminar la tarea de seleccionar características *a priori*, con un buen desempeño en la predicción de AMPs. En el presente trabajo, se propone un conjunto de datos para cada una de las actividades biológicas estudiadas, la más grande y diversa colección reportada a la fecha. Se usa el algoritmo de bosque aleatorio para construir modelos clásicos de clasificación. Se propone una red neuronal de múltiples capas con unidades recurrentes para construir modelos profundos. Los experimentos computacionales mostraron que los clasificadores propuestos en este trabajo, para cada actividad biológica, tienen un desempeño de comparable a superior respecto a los métodos del estado del arte. Por otra parte, se comparó el desempeño de modelos clásicos y profundos para el problema de clasificación de AMPs y las funciones atribuidas a estos, específicamente las funciones antibacteriana, antifúngica, antiparasitaria y antiviral. Los resultados mostraron que no existe una clara superioridad en el desempeño de un modelo con respecto al otro.

Palabras clave: aprendizaje de máquina, aprendizaje profundo, clasificación multiclase, péptidos antimicrobianos, selección de características

Abstract of the thesis presented by Sergio Alejandro Pinacho Castellanos as a partial requirement to obtain the Master of Science degree in Computer science.

Multi-class classification of antimicrobial peptides: a comparative approach

Abstract approved by:

Dr. Carlos Alberto Brizuela Rodríguez
Thesis Co-Director

Dr. César Raúl García Jacas
Thesis Co-Director

The increasing proliferation of infectious agents capable of resisting antimicrobial drugs constitutes a grave problem for human health worldwide. Antimicrobial peptides (AMPs) have gained popularity as promising candidates for use alone or in combination with current therapies to address antimicrobial resistance. To tackle the problem of discovering new AMPs with potential therapeutic use, computational models based on classical machine learning algorithms (known as shallow models) have been proposed. One problem with these models is that the computing cost of selecting a set of molecular descriptors to represent the data grows exponentially with the number of descriptors initially available for selection. Recently, computational models based on multi-layer neural networks (known as deep models) showed that the task of selecting features *a priori* could be eliminated, maintaining good performance in the prediction of AMPs. In this work, data sets were proposed for each of the studied biological activities, the largest and most diverse reported to date. The random forest algorithm was used to build shallow models. A multi-layer neural network with recurrent units was proposed to build deep models. Computational experiments showed that the proposed classifiers, for each biological activity, have a performance that goes from comparable to superior with respect to state of the art approaches. Additionally, shallow and deep models' performance was compared, addressing the problem of classification of AMPs and the functions attributed to them, specifically the antibacterial, antifungal, antiparasitic and antiviral functions. The results showed no clear superiority in the performance of the deep models as compared to the shallow models or vice versa.

Keywords: antimicrobial peptides, feature selection, deep learning, machine learning, multi-class clasification

Dedicatoria

Este trabajo de tesis está dedicado a mi esposa Milady y a mis hijos Dafne y Mateo, por ser mi principal motivación para continuar soñando despierto, y mi mayor fuerza para luchar por esos sueños.

Agradecimientos

Al Dr. César Raúl García Jacas, por contagiarme su dedicación al trabajo y su energía, por cada plática en el laboratorio y por cada llamada de atención que enriquecieron mi formación. Gracias por ser un excelente director de tesis y tenderme la mano como un amigo.

Al Dr. Carlos Alberto Brizuela Rodríguez, por ser más que un director de tesis, un amigo. Gracias, no solo por la guía y consejos para este trabajo, si no también por el apoyo moral y personal que siempre me brindó.

Al Dr. Israel Marck Martínez Pérez y al Dr. Sergio Andrés Águila Puentes, por ser parte de mi comité de tesis y orientarme siempre con sus preguntas y recomendaciones.

A todas aquellas personas que me apoyaron durante mi estancia en el posgrado; mis padres y hermanos, mis suegros, Mtra. Laura Rebeca, Dra. Rufina, Marcela, Karla, Liz, Gyna, Oli, María y Rafa, Rodrigo y Coral, Eric y Diana, Marlyne, y en general todos los papás y mamás de la generación 2017-2020 de la estancia infantil CICESE.

Al Centro de Investigación Científica y de Educación Superior de Ensenada por brindarme la oportunidad de ser parte de este posgrado, de igual manera, al excelente personal administrativo, Karina y Angelica, y a mis compañeros de generación por hacer grata mi estadía en esta institución.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de maestría. No. de becario: 808895. Así mismo, agradezco el apoyo otorgado mediante el proyecto FORDECYT 296737 ’Consorcio en Inteligencia Artificial’ para el desarrollo de la presente tesis.

Tabla de contenido

Página

Tabla de contenido (continuación)

Lista de figuras

Figura Página

Lista de figuras (continuación)

Lista de figuras (continuación)

Lista de tablas

Lista de tablas (continuación)

Lista de tablas (continuación)

Lista de tablas (continuación)

Capítulo 1. Introducción

La continua proliferación de agentes infecciosos capaces de resistir a los fármacos antimicrobianos que existen en el mercado, se ha convertido en una de las mayores preocupaciones para la salud pública mundial (World Health Organization *et al.*, 2014). Por lo anterior, pareciera que la era de los antibióticos convencionales está llegando a su final; esto ha llevado a muchas organizaciones y grupos de investigación a buscar nuevas estrategias para diseñar fármacos capaces de controlar este problema en crecimiento (Roca *et al.*, 2015).

Los péptidos antimicrobianos (AMPs, por sus siglas en inglés), son moléculas pequeñas que conforman el sistema inmune innato de organismos como mamíferos, plantas y hongos, son la primera línea de defensa contra microorganismos patógenos (Midura-Nowaczek y Markowska, 2014). Los AMPs han ganado popularidad al considerarse candidatos prometedores para usarse solos o en combinación con los antibióticos actuales para hacer frente a la resistencia antimicrobiana (Peters *et al.*, 2010; Lewies *et al.*, 2018), un ejemplo de su potencial uso se puede apreciar en el reciente péptido CSM5-k5 (Thappeta *et al.*, 2020) que combate a bacterias multi droga resistentes.

Un aspecto importante al proponer agentes antimicrobianos es la selectividad tóxica contra estos, lo cual es una característica distiguible en los AMPs debido a su interacción preferente con las células microbianas (Peters *et al.*, 2010). Adicionalmente, los AMPs exhiben un amplio espectro de actividades tales como antibacteriana, antifúngica, antiviral y antiparasitaria (Li *et al.*, 2012; Bahar y Ren, 2013; Midura-Nowaczek y Markowska, 2014). Más aún, los AMPs han mostrado baja predisposición a desarrollar resistencia por parte de los agentes infecciosos (Lewies *et al.*, 2019).

Desafortunadamente, existen diferentes barreras que impiden el desarrollo terapéutico de AMPs, entre ellas se pueden mencionar el alto costo de producción, susceptibilidad a degradarse por proteasas (Peters *et al.*, 2010; Aoki y Ueda, 2013) y, aunado a lo anterior, el espacio de búsqueda de péptidos con potencial uso terapéutico crece de manera exponencial respecto a la longitud de estos (e.g. considerando solamente la longitud mayor observada en AMPs se podrían obtener 20^{100} secuencias, lo cual es ampliamente superior al número total de átomos en el universo observable).

Tomando en cuenta las restricciones mencionadas, se han propuesto métodos que

ayudan a reducir el espacio de búsqueda y tener mayor certeza al probar péptidos candidatos, lo anterior, prediciendo si estos presentan la actividad deseada o carecen de ella (Torres y de la Fuente-Nunez, 2019). Uno de los métodos más utilizados para predecir actividad en péptidos es el modelo cuantitativo de relación estructura-actividad (QSAR, por sus siglas en inglés), el cual relaciona características físico-químicas, estructurales, conformacionales, entre otras de los péptidos con su actividad biológica (Roy *et al.*, 2015).

Gracias a la construcción de numerosas bases de datos de péptidos antimicrobianos y sus funciones biológicas conocidas; tales como APD3 (Wang *et al.*, 2015), ADAM (Lee *et al.*, 2015), CAMPR3 (Waghu *et al.*, 2016), DBAASP (Pirtskhalava *et al.*, 2015), DRAMP (Fan *et al.*, 2016), dbAMP (Jhong *et al.*, 2018) y StarpepDB (Aguilera-Mendoza *et al.*, 2019); así como al desarrollo de algoritmos de aprendizaje de máquina robustos, los modelos QSAR se potenciaron supliendo los métodos tradicionales para encontrar las relaciones estructura-actividad por modelos de aprendizaje de máquina (Fjell *et al.*, 2012), generalmente denominados clasificadores. El término clasificador se debe a que asignan una categoría (clase) a cada péptido de entrada, donde estas pueden ser, por ejemplo, AMP y non-AMP.

Para asistir a la investigación y descubrimiento de nuevos AMPs, se han desarrollado modelos computacionales basados en algoritmos de aprendizaje de máquina clásicos, tales como bosque aleatorio (Lin y Xu, 2016; Waghu *et al.*, 2016; Lin *et al.*, 2019; Chung *et al.*, 2019; Wei *et al.*, 2019), máquinas de soporte vectorial (Waghu *et al.*, 2016; Meher *et al.*, 2017; Gull *et al.*, 2019), *k* vecinos más cercanos (Xiao *et al.*, 2013; Wang *et al.*, 2017) y análisis discriminante lineal (Waghu *et al.*, 2016), que permiten identificar péptidos con actividad y aquellos que podrían carecer de esta.

Algunos modelos en la literatura reportan una exactitud en la clasificación de AMPs menor o igual al 95 % (Xiao *et al.*, 2013; Lin y Xu, 2016; Waghu *et al.*, 2016; Bhadra *et al.*, 2018; Veltri *et al.*, 2018; Chung *et al.*, 2019). Por otra parte, algunos estudios que han abordado el problema de predecir actividades específicas en AMPs (e.g. antibacteriana, antifúngica, antiviral, etcétera), alcanzan exactitudes iguales o inferiores al 84 % (Xiao *et al.*, 2013; Lin y Xu, 2016; Wang *et al.*, 2017; Meher *et al.*, 2017; Gull *et al.*, 2019; Lin *et al.*, 2019; Chung *et al.*, 2019; Wei *et al.*, 2019). Lo anterior, continúa dejando un espacio para mejorar y dar mayor certeza a la búsqueda de péptidos con

potencial uso terapeútico. En el mismo sentido, un problema al evaluar y comparar estos métodos es que no existe un conjunto de datos estandarizado, y las métricas reportadas podrían no ser del todo confiables.

Dado que la mayoría de los métodos del estado del arte caen dentro de la esfera de métodos QSAR, estos deben definir un conjunto de características cuantificables (descriptores moleculares) en los péptidos que les permitan encontrar los patrones de relación estructura-actividad (Roy *et al.*, 2015). Hoy en día el orden en número de descriptores moleculares que pueden calcularse es de miles, por ello, la selección de un conjunto adecuado para predecir la actividad deseada se torna un problema difícil (Goodarzi *et al.*, 2012).

Recientemente, Veltri *et al.* (2018) mostraron que las redes neuronales profundas podrían eliminar la tarea de seleccionar características *a priori*, con un buen desempeño en la predicción de AMPs, con esto se podría reducir la complejidad en la construcción de un clasificador. Así mismo, tanto Hamid y Friedberg (2019), como Su *et al.* (2019), alcanzan exactitudes en la clasificación de AMPs comparables a las obtenidas por los demás métodos reportados en la literatura. Lo anteiror, con métodos basados en redes neuronales profundas sin la necesidad de definir un conjunto de descriptores moleculares.

Lo expuesto hasta el momento, abre la interrogante sobre qué tipo de algoritmo de aprendizaje de máquina, de tipo clásico o profundo, es mejor para construir modelos de clasificación de péptidos antimicrobianos y sus funciones asociadas, además, bajo qué circunstancias es mejor uno del otro. Tomando como punto de partida la pregunta anterior, en la siguiente sección se exponen los objetivos establecidos para este trabajo de investigación.

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar modelos de aprendizaje de máquina basados en algoritmos clásicos y profundos para reconocer péptidos con actividad antimicrobiana y sus funciones específicas de antibacterial, antifúngica, antiparasitaria y antiviral.

1.1.1.1. Objetivos específicos

- Construir un conjunto de datos etiquetados de AMPs y sus funciones conocidas.
- Identificar los descriptores moleculares que describen mejor a los AMPs según su función.
- Proponer modelos de aprendizaje de máquina clásicos para identificar AMPs y sus funciones de antibacterial, antifúngica, antiparasitaria y antiviral
- Proponer modelos de aprendizaje profundo para reconocer AMPs y las funciones biológicas estudiadas.
- Comparar el rendimiento de modelos clásicos y profundos en la clasificación de péptidos antimicrobianos.
- Comparar el desempeño de los modelos propuestos en este trabajo con los métodos del estado del arte.

1.2. Metodología propuesta

La metodología general propuesta para alcanzar cada uno de los objetivos planteados se muestra en la Figura 1. Primero, se construyeron los conjuntos de datos y se llevó a cabo un proceso de filtrado de los mismos. Enseguida, se propuso un método para dividir los conjuntos en subconjuntos de entrenamiento y prueba. Posteriormente, se llevó a cabo un proceso de selección de características. Después, se construyeron modelos clásicos de clasificación y profundos. Más adelante, se evaluaron los modelos y se comparó su rendimiento, así mismo, se contrastó su desempeño con los métodos del estado del arte. Por último, con los modelos mejor evaluados, se predijeron péptidos con potencial actividad antimicrobiana como ejemplo de aplicación.

1.3. Organización de la tesis

El presente documento está organizado en cinco capítulos descritos a continuación. El Capítulo 1, introduce y motiva esta investigación, además de delimitar los alcances de la misma. El Capítulo 2, proporciona los fundamentos teóricos necesarios para entender lo desarrollado en este trabajo. El Capítulo 3, detalla la metodología empleada

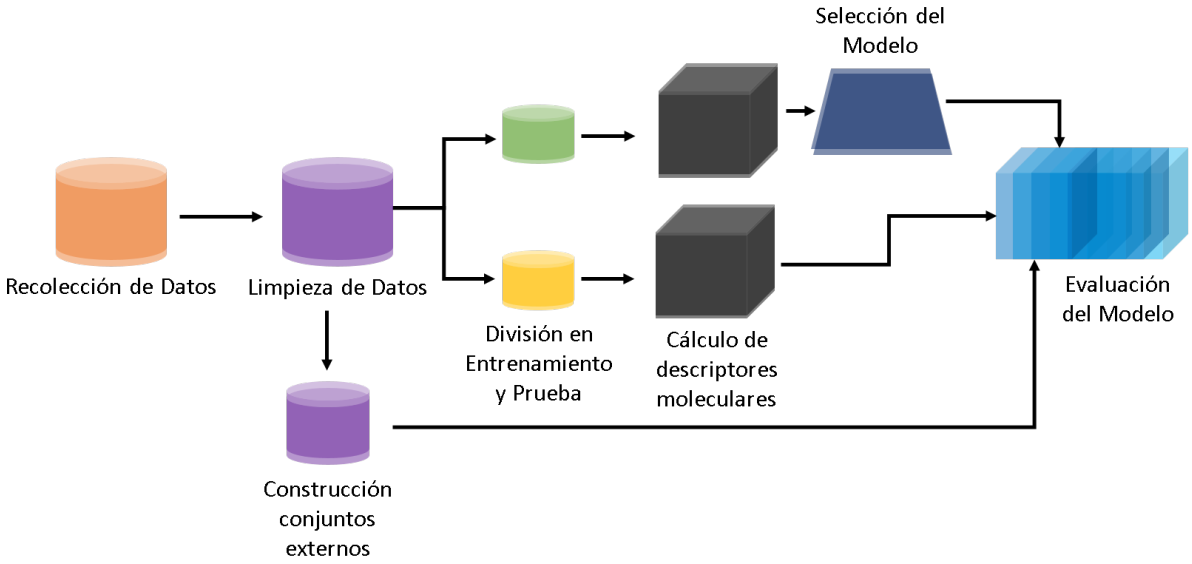

Figura 1. Metodología general propuesta.

para cumplir con los objetivos estipulados. El Capítulo 4, expone los resultados obtenidos al aplicar la metodología propuesta. El Capítulo 5, discute los resultados obtenidos, así mismo, presenta las conclusiones a las que se llegaron con esta investigación.

Adicionalmente, se añaden cuatro anexos como material suplementario. El Anexo A, contiene los resultados en el número de descriptores moleculares seleccionado en cada proceso llevado a cabo, de igual manera, brinda un listado de los descriptores elegidos para entrenar los clasificadores mejor evaluados para cada actividad estudiada. El Anexo B, ofrece la relación de péptidos considerados para predecir su potencial actividad antimicrobiana. El Anexo C, contiene los resultados obtenidos al construir un modelo por ensamble. Por último, el Anexo D presenta los valores del error cuadrático medio obtenido por cada clasificador.

Capítulo 2. Marco Teórico

El presente capítulo expone los conceptos teóricos básicos que necesita el lector para entender lo desarrollado en este trabajo de investigación. Se presentan inicialmente los temas referentes a aprendizaje de máquina y, posteriormente, lo requerido para conocer el problema de clasificación de péptidos antimicrobianos.

2.1. Aprendizaje de máquina

Un *algoritmo de aprendizaje de máquina* es un algoritmo capaz de aprender de un conjunto de datos (frecuentemente llamado conjunto de entrenamiento) (Goodfellow *et al.*, 2016). En el mismo sentido, Bishop (2006) define el aprendizaje como la capacidad de mejorar el rendimiento en tareas futuras tras hacer observaciones de lo conocido. En otras palabras, se dice que un programa de computadora aprende de una experiencia E respecto a algún tipo de tarea T y con una medida de desempeño P, si su desempeño en la tarea T, bajo la medición P, mejora con la experiencia E (Mitchel, 1997).

El desafío central en tareas de aprendizaje es que los algoritmos funcionen bien en datos que no fueron observados durante el proceso de aprendizaje, esta habilidad es denominada *generalización*. Para medir la capacidad de generalización de un modelo de aprendizaje, debemos definir un criterio cuantificable de su desempeño (la medida P), la cual puede ser obtenida usando un conjunto de datos diferente al conjunto de entrenamiento (regularmente denominado conjunto de prueba) (Goodfellow *et al.*, 2016), o bien, usando diferentes técnicas de evaluación sobre el conjunto de entrenamiento como redistribución, reserva, reserva repetida, validación cruzada, dejar uno fuera, entre otros (Kuncheva, 2014).

Algunas medidas para evaluar la generalización de un modelo de aprendizaje son la sensibilidad (SN, también conocida como recall), especificidad (SP), exactitud (ACC) y coeficiente de correlación de Matthews (MCC), las cuales se definen como:

$$SN = \frac{TP}{TP + FN}, \tag{1}$$

$$SP = \frac{TN}{FP + TN}, \tag{2}$$

$$ACC = \frac{TP + TN}{TP + FN + TN + FP}, \tag{3}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}, \tag{4}$$

donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos.

Ahora bien, centrándonos en la tarea T, estas son descritas usalmente en términos de cómo el algoritmo de aprendizaje procesa los ejemplos de la experiencia E. Comúnmente, un ejemplo es representado como un vector $\mathbf{x} \in \mathbb{R}^n$, donde cada uno de sus componentes x_i representa alguna característica medible sobre un objeto o evento (Goodfellow *et al.*, 2016).

Tomando en cuenta la experencia E, la tarea T puede categorizarse en dos grupos según en el tipo de experiencia de la que disponga durante el proceso de aprendizaje. Si para cada ejemplo $\mathbf{x}$ en el conjunto de datos, existe asociado a este una etiqueta u objetivo 'y' que describe el valor real del ejemplo, entonces se categoriza como una tarea de aprendizaje supervisado, si estas etiquetas no existen se considera una tarea de aprendizaje no supervisado (Donalek, 2011).

Considerando lo descrito en el párrafo anterior, a continuación se presentan las tareas de aprendizaje abordadas en este trabajo y los algoritmos empleados para resolverlas.

2.1.1. Clasificación

Esta es un tipo de tarea supervisada en la cual el algoritmo de aprendizaje trata de especificar para cada ejemplo $\mathbf{x}$ una única etiqueta 'y' de un conjunto disjunto de etiquetas $Y = \{y_1, ..., y_k\}$. Para resolver esta tarea, usualmente, el algoritmo de aprendizaje busca producir una función $f : \mathbf{x} \rightarrow Y$. Si $k = 2$ la tarea es denominada

clasificación binaria, cuando $k > 2$ la tarea se conoce como clasificación multiclase (Tsoumakas y Katakis, 2007; Goodfellow *et al.*, 2016).

Debido a la complejidad intrínseca de un problema de clasificación multiclase, la posibilidad de cometer errores en la clasificación es mayor que en el caso binario (Lorena *et al.*, 2008), por ello, se han adoptado diferentes técnicas para tratar problemas de esta índole, tales como adaptaciones a los algoritmos de clasificación y técnicas de descomposición (Aly, 2005), estas últimas reducen un problema con k categorías a k problemas binarios, posteriormente usan alguna técnica para combinar sus resultados y agregarlos como uno solo.

Enseguida, se presentan algunos ejemplos de algoritmos utilizados para resolver tareas de clasificación.

2.1.1.1. Árboles de decisión y bosque aleatorio

Según Russell y Norvig (2016) un árbol de decisión (DT, por sus siglas en inglés), para un problema de clasificación binaria, representa una función h que toma como entrada un vector de atributos $\mathbf{x}$ y regresa un único valor de salida 'y'. La salida de un DT está dada por una serie de decisiones, cada nodo interno representa una decisión sobre el valor de un atributo $x_i \in \mathbf{x}$, las ramas de los nodos están etiquetadas con los posibles valores de los atributos y los nodos hojas representan el valor $y \in Y$ retornado por la función.

El Algoritmo 1 describe el proceso de aprendizaje para un DT. En las líneas 1 a 9 se establecen condiciones de parada que verifican si no hay elementos aún por asignar a alguna categoría, si todos los elementos pertenecen a la misma categoría o si todos los atributos han sido asignados a alguna rama del árbol. La función *PLURALITY-VALUE*, selecciona el valor de salida más común entre un conjunto de ejemplos. En las líneas 11 a 17, se construye el árbol de forma recursiva seleccionando en cada nodo el atributo que genera la mejor decisión para separar las categorias. La función *IMPORTANCE*, define la calidad de la decisión tomada en los nodos. Se han propuesto diferentes funciones de importancia basadas en criterios de entropía, índice Gini, error de clasificación, ganancia de información, relación de ganancia y criterio binario (Singh y Gupta, 2014).

Algoritmo 1: DECISION-TREE-LEARNING

Input: examples,attributes,parent_examples
Output: a tree

if *examples is empty* **then**
  return *PLURALITY-VALUE(parent_examples)*
end
else if *all examples have the same classification* **then**
  return *the classification*
end
else if *attributes is empty* **then**
  return *PLURALITY-VALUE(examples)*
end
else
  $A \leftarrow argmax_{a \in attributes}$IMPORTANCE(a,examples)
  tree ← a new decision tree with root test A
  foreach *value* v_k *of* A **do**
    $exs \leftarrow \{e : e \in examples \text{ and } e.A = v_k\}$
    subtree ← DECISION-TREE-LEARNING(exs,attributes - A,examples)
    add a branch to *tree* with label (A = v_k) and subtree *subtree*
  end
end
return *tree*

Para un problema de clasificación binaria, el criterio de optimización de un DT está definido por la siguiente expresión (Burkov, 2019) la cual debe ser maximizada:

$$\frac{1}{N}\sum_{i=1}^{N}[y_i \, ln(h(x_i)) + (1-y_i) \, ln(1-h(x_i))], \tag{5}$$

donde N es el número de ejemplos en el conjunto de entrenamiento, h es un DT y y_i es la etiqueta asociada al ejemplo i.

Una vez conceptualizado el algoritmo de aprendizaje de un DT, se puede introducir el concepto de bosque. Tomando la definición de Breiman (2001), un bosque aleatorio (RF, por sus siglas en inglés), es un clasificador que consiste de una colección de clasificadores estructurados como árboles, dados por $\{h(X, \Phi_k), k = 1, ..., r\}$ donde h es un DT, X un conjunto de datos con cada ejemplo en el conjunto con m variables de entrada, Φ_k son vectores aleatorios independientes e idénticamente distribuidos y r es el número máximo de árboles en la colección. Cada árbol emite un voto unitario para obtener la clase más popular a la que pertenece cada entrada $\mathbf{x} \in X$.

El término aleatorio en el algoritmo RF alude a dos cosas, en principio, a Φ_k, un vector que representa una selección aleatoria con reemplazo de ejemplos en X, los cuales son usados para construir el k-ésimo DT (proceso conocido como *bagging*). En segundo lugar, se debe a que en la construcción del DT, en cada nodo, en lugar de usar las m variables de entrada para tomar una decisión, se usa solo un número aleatorio de estas (Breiman, 2001).

2.1.1.2. Redes neuronales artificiales

Según Goodfellow *et al.* (2016) los perceptrones multicapa (MLPs, por sus siglas en inglés) son los modelos de aprendizaje profundo por excelencia. Los MLPs también son conocidos como redes neuronales profundas (DNNs), esto debido a que pueden representarse como una composición de diferentes funciones y, a su vez, estas composiciones, se pueden representar usando un grafo dirigido acíclico (similar a una red). El término profundo, hace referencia a la longitud de la cadena de composiciones formada por diferentes funciones. La Figura 2 muestra la representación gráfica de una red neuronal con cuatro funciones, cada función toma la salida z_i de la función que la precede.

El objetivo de una DNN es aproximar alguna función f^*. Por ejemplo, dado un problema de clasificación y una función $f^* : \mathbf{x} \rightarrow Y$, que asigna a cada entrada $\mathbf{x}$ una etiqueta 'y', una DNN define una función $f_W : \mathbf{x} \rightarrow Y$ que aprende los parámetros W que mejor aproximan a la función f^* (Goodfellow *et al.*, 2016). Para encontrar los parámetros W, se propuso el algoritmo de retropropagación (Rumelhart *et al.*, 1986) el cual utiliza métodos de optimización como el descenso de gradiente para minimizar una función de costos, la cual puede definirse según Bonaccorso (2020) como :

$$J(X, Y; W) = \sum_{i=0}^{N} L(x_i, y_i; W), \tag{6}$$

donde N es el número de ejemplos en el conjunto de entrenamiento. La función L, es conocida como función de pérdida, esta puede variar según el tipo de función que se desee aproximar, y trata de medir las discrepancias entre las salidas de la red neuronal y las salidas reales esperadas (Bonaccorso, 2020). Desde otra perspectiva, una red

$$f_W(\boldsymbol{x}) = f^{(4)}\left(f_{W_3}^{(3)}\left(f_{W_2}^{(2)}\left(f_{W_1}^{(1)}(\boldsymbol{x})\right)\right)\right) = y$$

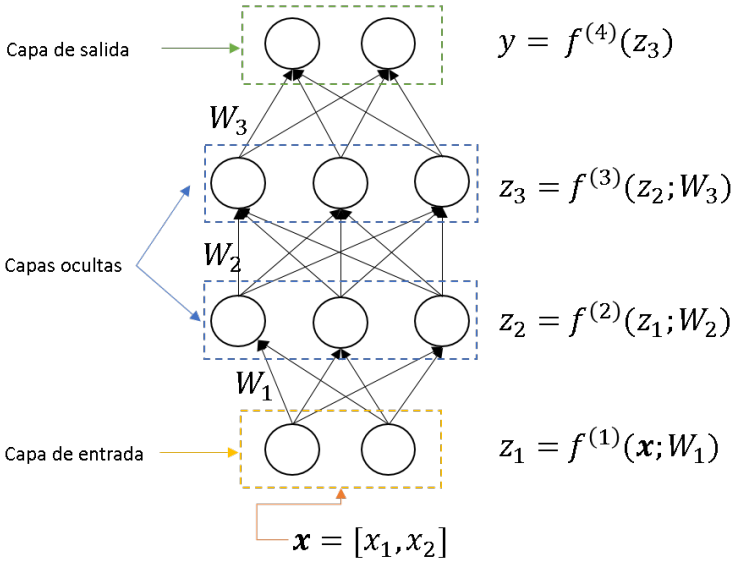

Figura 2. Representación de una red neuronal profunda. $f^{(1)}$ es la primer capa, también llamada capa de entrada; $f^{(2)}$ y $f^{(3)}$, son la segunda y tercera capa, también llamadas capas ocultas; y $f^{(4)}$ es la última capa de la red, conocida como la capa de salida.

neuronal busca encontrar los parámetros W que satisfagan la siguiente ecuación:

$$W^* = argmin_W J(X, Y; W). \tag{7}$$

Además del modelo general de MLPs, en la literatura se han propuesto diferentes arquitecturas de redes neuronales como las redes neuronales convolucionales (Krizhevsky *et al.*, 2012), y las redes neuronales recurrentes (Rumelhart *et al.*, 1986), siendo estas últimas las que han mostrado mejor rendimiento en problemas que involucran datos estructurados secuencialmente (Graves *et al.*, 2013; Sutskever *et al.*, 2014; Bahdanau *et al.*, 2014). A continuación, hacemos una breve descripción de las redes neuronales recurrentes.

2.1.1.3. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) atacan uno de los problemas más comunes en los MLPs, la longitud variable de la entrada. Para lidiar con este problema, las RNNs utilizan una función oculta recurrente h_t sobre los estados t de la entrada, es decir,

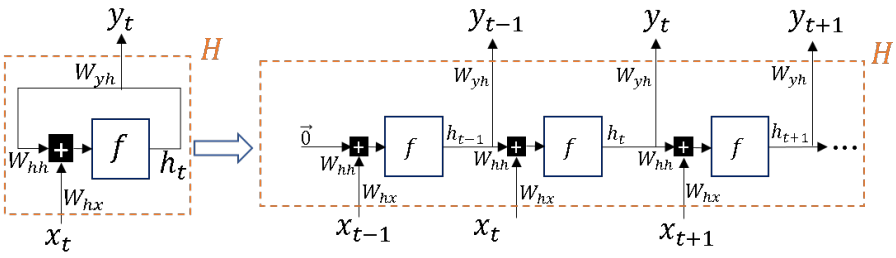

Figura 3. Representación gráfica de una red neuronal recurrente en su estado abstracto y posteriormente extendida en el tiempo. Figura adaptada de (Shrestha y Mahmood, 2019).

en lugar de considerar una secuencia de entrada $\mathbf{x} = [x_1, ..., x_r]$, donde r es variable, recibe como entrada cada componente x_t de esta; y cada estado h_t depende del estado anterior, como se aprecia en la Figura 3. Esta función de recurrencia esta definida formalmente como:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1}) \tag{8}$$

donde f es comúnmente la función logística o la función tangente hiperbólica. W_{hx} y W_{hh} son parámetros a optimizar durante el proceso de aprendizaje, y, frecuentemente, h_0 es un vector de ceros (Chung *et al.*, 2014).

La salida de una unidad recurrente, a diferencia de las unidades simples en MLPs, se calcula basada en las funciones ocultas h_t, normalmente implementada como:

$$y_t = f(W_{yh}h_t). \tag{9}$$

Uno de los principales problemas observados en las RNNs es la dificultad para capturar dependencias a largo plazo en las secuencias, debido a que los gradientes tienden a desvanecerse durante el proceso de aprendizaje (Bengio *et al.*, 1994). Para abordar este problema se han propuesto arquitecturas de unidades recurrentes más complejas, a continuación se describen las arquitecturas más utilizadas.

Unidades de memoria a largo corto plazo

A diferencia de las unidades simples de RNN, una unidad H de memoria a largo corto plazo (LSTM) (Hochreiter y Schmidhuber, 1997) mantiene una memoria en ca-

da tiempo t de la entrada. Intuitivamente, las unidades LSTM identifican información relevante en las etapas iniciales de la entrada y acarrean dicha información hacia las etapas finales, los estados h_t de una unidad recurrente H son calculados según Chung *et al.* (2014) como:

$$h_t^H = o_t^H \; tanh(c_t^H), \tag{10}$$

donde o_t^H funciona como una compuerta que modula el contenido de la memoria, y se calcula como sigue:

$$o_t^H = \sigma(W_{ox} x_t \; + \; W_{oh} h_{t-1} \; + V_o c_t), \tag{11}$$

donde σ es la función logística y V_o es una matriz diagonal.

La celda de memoria c_t^H se actualiza parcialmente olvidando la información de la celda existente y se añade una nueva memoria candidata $\tilde{c}_t^H$ como sigue:

$$c_t^H = f_t^H c_{t-1}^H \; + \; i_t^H \tilde{c}_t^H, \tag{12}$$

donde el contenido de la memoria candidata $\tilde{c}_t^H$ es calculado como:

$$\tilde{c}_t^H = tanh(W_{cx} x_t \; + \; W_{ch} h_{t-1}). \tag{13}$$

La actualización de la celda de memoria está modulada por una función f_t^H que simula una compuerta que olvida información de la memoria actual, y una función i_t^H modula la información que es agregada desde la memoria candidata. Estas funciones están definidas según:

$$f_t^H = \sigma(W_{fx} x_t \; + \; W_{fh} h_{t-1} \; + V_f c_{t-1}), \tag{14}$$

$$i_t^H = \sigma(W_{ix} x_t \; + \; W_{ih} h_{t-1} \; + V_i c_{t-1}), \tag{15}$$

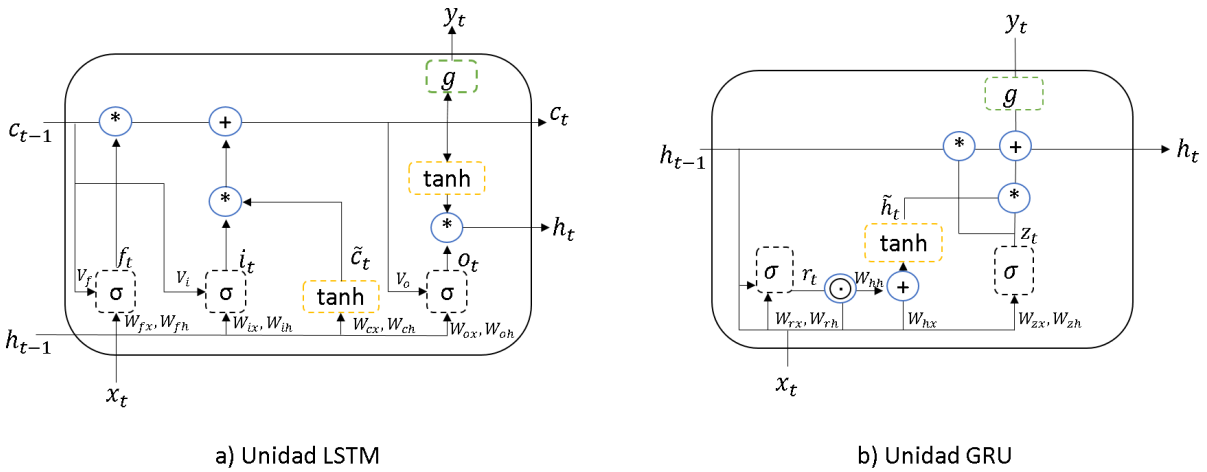

Figura 4. a) Representación gráfica de una unidad LSTM. b) Representación gráfica de una unidad GRU. Figura adaptada de (Ng.et al., s.f.).

donde V_f y V_i son matrices diagonales. La Figura 4a muestra, de forma gráfica, la composición de una unidad LSTM.

Unidades recurrentes con compuertas

Una unidad recurrente con compuertas *H* (GRU) (Cho *et al.*, 2014), contiene unidades que manipulan el flujo de información, similar a las unidades LSTM, la diferencia es que no utilizan una celda de memoria. Los estados h_t^H son calculados según Chung *et al.* (2014) como:

$$h_t^H = (1 - z_t^H)h_{t-1}^H + z_t^H \tilde{h}_t^H, \tag{16}$$

donde z_t^H es una función que controla la información a actualizar, calculada como:

$$z_t^H = \sigma(W_{zx}x_t + W_{zh}h_{t-1}). \tag{17}$$

Este procedimiento toma una suma lineal entre el estado existente y un nuevo estado candidato, otra diferencia con las unidades LSTM es que en este caso no se controla el grado de actualización del estado actual. El estado candidato $\tilde{h}_t^H$ es calculado con la siguiente expresión:

$$\tilde{h}_t^H = tanh(W_{hx}x_t + W_{hh}(r_t \odot h_{t-1})), \tag{18}$$

donde r_t es una función de reconfiguración del estado, si sus valores son cercanos a cero, esta función actúa como si olvidará la información del estado actual, y se agrega la información del estado candidato, el símbolo $\odot$ representa una multiplicación elemento a elemento entre el vector devuelto por la función r_t y el vector representante del estado h_{t-1}. La siguiente expresión calcula la función r_t:

$$r_t^H = \sigma(W_{rx}x_t + W_{rh}h_{t-1}). \tag{19}$$

La Figura 4b muestra en forma gráfica una unidad GRU.

2.1.2. Agrupamiento

De acuerdo con Bishop (2006) este tipo de tarea de aprendizaje no supervisado, dado un conjunto de datos $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, con N igual al número de ejemplos en el conjunto de datos, y cada ejemplo es una variable euclidiana D-dimensional, $\mathbf{x}_i = \{x_1,...,x_D\}$, tiene como objetivo dividir el conjunto X en un número K de grupos. Intuitivamente, se puede pensar en cada grupo $C_i \subset X$, con $i = 1, ..., K$, como un subconjunto de datos para el cual se cumple lo siguiente: dada una función de distancia δ, la distancia $\delta(\mathbf{x}_i, \mathbf{x}_j)$, $\forall \mathbf{x}_i, \mathbf{x}_j \in C_i$, es menor a la distancia $\delta(\mathbf{x}_i, \mathbf{x}_r)$, $\forall \mathbf{x}_i \in C_i$ y $\forall \mathbf{x}_r \notin C_i$.

Se han desarrollado diferentes algoritmos para solucionar problemas de agrupamiento, uno de ellos es el algoritmo de agrupamiento por maximización de la esperanza el cual se presenta a continuación.

2.1.2.1. Algoritmo de agrupamiento por maximización de la esperanza

Según Jin y Han (2010), el algoritmo de agrupamiento por maximización de la esperanza (EMC), se basa en el algoritmo de maximización de la esperanza (EM) propuesto por Dempster *et al.* (1977). Donde sobre un conjunto de datos X define un modelo de mezclas Gausianas como un conjunto de K distribuciones de probabilidad, y cada distribución representa un grupo $C_i \subset X$, con $i = 1, ..., K$. Cada ejemplo en el conjunto

de datos es asignado con cierta probabilidad a cada grupo. El algoritmo EMC, trabaja de la siguiente manera:

1. Propone parámetros iniciales: media y desviación estándar (si se usa un modelo de distribución normal).
2. Optimiza los parámetros iterativamente con los pasos de esperanza E y maximización M. En el paso E, calcula la probabilidad de pertenencia de cada ejemplo $\mathbf{x} \in X$ para cada grupo C_i, basado en los valores de los parámetros iniciales. En el paso M, recalcula los parámetros basados en las probabilidades de pertenencia calculados en el paso E.
3. Asigna a cada ejemplo $\mathbf{x} \in X$ al grupo C_i para el que se tiene la mayor probabilidad de pertenencia.

Aunque se han analizado dos tareas de aprendizaje, este trabajo de investigación tiene como objetivo comparar el desempeño de algoritmos de clasificación conocidos como clásicos, como el algoritmo RF, y profundos, como las DNNs. Para ello, se tomó el problema de clasificación multiclase de péptidos antimicrobianos, como ejercicio computacional, para comparar el rendimiento de estos algoritmos. En la siguiente sección se analizan los conceptos biológicos sobre el problema mencionado.

2.2. Clasificación multiclase de péptidos antimicrobianos

Primero se debe introducir el concepto de oligopéptido, sin embargo, antes de eso, es imprescindible conocer los bloques estructurales que los conforman, los aminoácidos. De la definición de Sung (2009), los aminoácidos son moléculas compuestas principalmente por los siguientes elementos:

1. Un grupo amino (grupo $-NH_2$).
2. Un grupo carboxilo (grupo $-COOH$).
3. Un grupo R (cadena lateral), el cual determina el tipo de aminoácido.

Los tres grupos están atados a un solo átomo de carbono, llamado carbono-α. La mayoría de las proteínas que componen a los seres vivos, están principalmente compuestas por 20 aminoácidos, diferenciados por el grupo R, los cuales son denominados aminoácidos estándar o naturales (Jiménez y Merchant, 2003).

Ahora bien, los oligopéptidos son polímeros formados por aminoácidos unidos a través de enlaces peptídicos, donde un enlace peptídico se forma cuando el grupo carboxilo de un aminoácido reacciona con el grupo amino de otro (McKee y McKee, 2009).

Según McKee y McKee (2009), existen varios niveles de organización estructural de los oligopéptidos. La *estructura primaria*, constituída por la secuencia de aminoácidos. La *estructura secundaria*, formada por las disposiciones de los aminoácidos adyacentes al plegarse la cadena polipeptídica, siendo las más comunes la hélice α y las láminas plegadas β. La *estructura terciaria* está dada por la forma tridimencional global que adopta un polipéptido. Las estructuras formadas por la unión de dos o más cadenas polipeptídicas conforman *la estructura cuaternaria*.

2.2.1. Péptidos antimicrobianos

Los péptidos antimicrobianos (AMPs, por sus siglas en inglés) son oligopéptidos de longitud variable, normalmente entre 5 y 100 aminoácidos (Bahar y Ren, 2013). Estos péptidos pueden organizarse en una estructura secundaria en la que los aminoácidos hidrofílicos e hidrofóbicos se segregan espacialmente, esto dota a los AMPs de una carga neta catiónica (Matsuzaki, 2019). Los péptidos con estas propiedades tienden a ser anfifílicos, es decir, son solubles en ambientes acuosos, pero también pueden interactuar con ambientes lipídicos como las membranas (Zasloff, 2002).

Considerando lo anterior, los AMPs exhiben una clara diferencia en el diseño de su membrana respecto a la de microbios y organismos multicelulares como plantas y animales. Las capas externas de las membranas de microbios están principalmente formadas por grupos de fosfolípidos con carga negativa, y las capas externas de membranas de plantas y animales están compuestas, principalmente, por lípidos sin carga neta (Zasloff, 2002). Debido a estas diferencias, los AMPs presentan los siguientes mecanismos de acción (Brogden, 2005):

1. *Atracción*: los péptidos son atraídos a las superfecies de los microbios, principalmente mediante uniones electrostáticas.

2. *Acoplamiento*: Una vez cerca de la superficie microbiana, los péptidos interacúan con las barreras fosfolípidas para ingresar a la membrana citoplasmática.

3. *Inserción del péptido y permeabilidad de la membrana*: A altas concentraciones, los péptidos se orientan perpendicularmente, se insertan en la bicapa lipídica y se forma un poro transmembranal (denominado estado), es decir, se perfora la membrana del objetivo. Si se desea conocer los modelos propuestos para explicar la permeabilización de las membranas se refiere al lector al trabajo original de Brogden (2005).

Los AMPs han mostrado un amplio espectro de actividades antimicrobianas tales como antifúngica, antibacteriana, antiparasitaria y antiviral, (Midura-Nowaczek y Markowska, 2014). Identificar la actividad específica de un AMP se puede modelar como un problema de clasificación multiclase (ver sección 2.1.1), dado un conjunto X de AMPs, se trata de encontrar una función f que asigne a cada ejemplo $\mathbf{x}_i \in X$ una etiqueta $y \in$ {antibacteriano, antifúngico, antiviral, ...}.

Una de las principales herramientas utilizada para identificar actividades en proteínas y péptidos es conocida como cribado virtual, la siguiente sección expone el fucionamiento de dicha herramienta.

2.2.2. Cribado virtual (*virtual screening*)

Según Roy *et al.* (2015), el cribado virtual (VS) es una técnica para identificar nuevas moléculas bioactivas en grandes bases de datos de compuestos químicos mediante el empleo de métodos computacionales y el conocimiento basado en la estructura de los compuestos o basado en ligandos bioactivos conocidos. Los enfoques basados en ligandos utilizan datos de estructura-actividad de un conjunto de actividades conocidas, para reconocer nuevos compuestos candidatos para su evaluación experimental (Villoutreix *et al.*, 2007).

Uno de los métodos basados en ligandos más aceptados para predecir actividad en péptidos es conocido como QSAR (*Quantitative Structure-Activity Relationship*), este

permite desarrollar modelos matématicos para identificar con presición actividades biológicas en compuestos desconocidos (Roy *et al.*, 2015).

Para encontrar estas relaciones de estuctura-actividad, el modelo QSAR se basa en características cuantificables de la estructura primaria y características físico-químicas de los péptidos, en otras palabras, los modelos QSAR describen la relación entre descriptores como variables de entrada y la actividad biológica como variable de salida.

2.2.2.1. Descriptores moleculares

El método QSAR utiliza información química estructural cuantificable de los péptidos, estas propiedades son conocidas como *descriptores moleculares*. Los descriptores moleculares pueden ser definidos como representaciones matemáticas de propiedades moleculares, generados por un algoritmo. Estos valores numéricos son usados para describir cuantitativamente la información física y química de una molécula (Chandrasekaran *et al.*, 2018).

Aunque los descriptores moleculares pueden obtenerse de diferentes niveles estructurales del péptido, en esta investigación nos centramos en descriptores que se obtienen de la estructura primaria de estos. A continuación se describen los descriptores utilizados.

Pseudo composición de aminoácidos (PseAAC)

El algoritmo de PseAAC codifica la secuencia de un péptido de la siguiente manera (Chou, 2001):

Sea $P = R_1, \ldots, R_L$ una secuencia de longitud L, donde R_i es el i-ésimo aminoácido en la secuencia, con $i = 1, \ldots, L$; el algoritmo PseAAC asigna a P un vector de longitud $20 + \lambda$, con $\lambda < L$.

Sea $P' = p_1, \ldots, p_{20}, p_{20+1}, \ldots, p_{20+\lambda}$ la secuencia codificada con el algoritmo PseAAC, cada p_u es calculado como sigue,

$$P_u = \begin{cases} \frac{f_u}{\sum_{i=1}^{20} f_i + \omega \sum_{k=1}^{\lambda} \tau_k}, & 1 \leq u \leq 20 \\ \frac{\omega \tau_{u-20}}{\sum_{i=1}^{20} f_i + \omega \sum_{k=1}^{\lambda} \tau_k}, & 21 \leq u \leq 20 + \lambda \end{cases} \tag{20}$$

donde f_u, con $u = 1, 2, \ldots, 20$, es una función que describe la frecuencia de ocurrencia de los 20 aminoácidos naturales en P, ω es un factor de ponderación y τ_k es el k-ésimo factor que refleja la correlación en el orden de la secuencia de k aminoácidos contiguos según,

$$\tau_k = \frac{1}{(L-k)} \sum_{i=1}^{L-k} J_{i,i+k}, k < L \tag{21}$$

con

$$J_{i,i+k} = \frac{1}{\gamma} \sum_{q=1}^{\gamma} \left[\phi_q (R_{i+k}) - \phi_q (R_i) \right]^2, \tag{22}$$

donde $\phi_q(R_i)$ es la q-ésima función del aminoácido R_i (e.g. hidrofobicidad, hidrofilia o masa de la cadena lateral) y γ el número total de funciones.

Descriptores moleculares de ProtDCal

ProtDCal (Romero-Molina *et al.*, 2019), es un software que transforma la información estructural de proteínas a descriptores moleculares usando diferentes operadores de agregación.

Para usar la herramienta ProtDCal se debe configurar una serie de parámetros que le indican al software qué tipo de descriptores calcular, primero, se debe seleccionar el o los tipos de índices que se usarán para codificar los aminoácidos, ProtDCal proporciona tres tipos de índices: termodinámicos, topográficos y basados en propiedades (propiedades físico-químicas y estructurales).

De igual manera, se debe indicar si los índices serán modificados, o no, considerando las vecindades de cada aminoácido, ProtDCal provee cinco operadores de modificación, también denominados operadores de vecindad (OV), estos son: autocorrela-

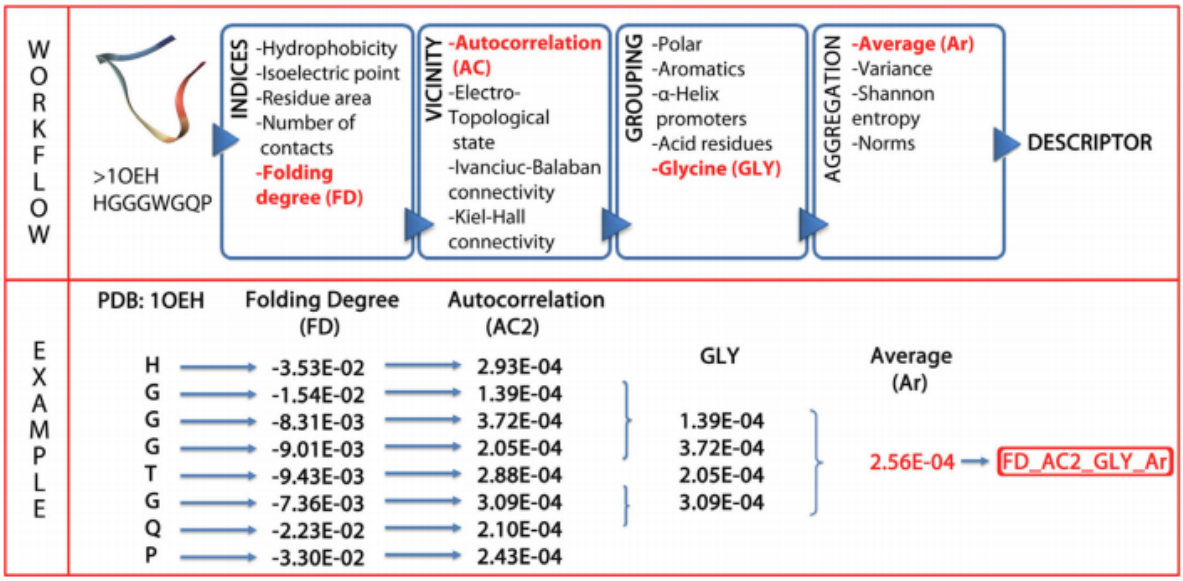

Figura 5. Flujo de trabajo de ProtDCal y ejemplo en el cálculo de un descriptor, tomado del artículo original (Romero-Molinaet al., 2019).

ción, gravitacional, Kier Hall, Ivanciuc Balaban y estado electrotopológico (Todeschini y Consonni, 2008). Algunos OVs permiten modificar el punto de corte en la distancia topológica de los aminoácidos, utilizando el parámetro *cutoff*.

Adicionalmente, se debe elegir un tipo de grupo por el cual los valores de los índices serán agregados, ProtDCal implementa tres tipos de grupos, los cuales están basados en residuos (aminoácidos), basados en propiedades y topográficos. Por último, es necesario escoger los operadores de agregación (OAs) que combinarán los valores de los grupos a un solo descriptor molecular, algunos ejemplos de estos operadores suministrados en ProtDCal son la suma, la media aritmética, la varianza, entre otros. La Figura 5 muestra el fujo de trabajo de ProtDCal y un ejemplo del cálculo de un descriptor.

Considerando que algunas representaciones basadas en descriptores moleculares de los péptidos pueden generar miles de estos, es común realizar un proceso de selección de características para encontrar un subconjuntos de descriptores que mejoren el rendimiento en la predicción de la actividad asociada a un péptido. Por esto, en la siguiente sección se describen los métodos de selección de características usados en este trabajo.

2.2.2.2. Selección de características

Un proceso de selección de características permite entender mejor los datos, reducir los requerimientos de cómputo y mejorar el rendimiento de un predictor, esto último reduciendo el ruido generado por características irrelevantes (Chandrashekar y Sahin, 2014). Una característica (descriptor molecular) $d_i \in D$, se dice que es fuertemente relevante para un conjunto de datos X, si existen dos ejemplos $\mathbf{x}, \mathbf{z} \in X$ que solo difieren en un valor d_i y, dado un objetivo de clasificación $c : X \rightarrow Y$, donde Y es un conjunto disjunto de etiquetas, $c(\mathbf{x}) \neq c(\mathbf{z})$. En otras palabras, se puede clasificar a $\mathbf{x}$ y $\mathbf{z}$ solamente usando d_i (Molina *et al.*, 2002).

Sea D el conjunto original de características, con $|D| = n$, y sea $J(D')$ una medida de evaluación a optimizar, definida como $J : D' \subseteq D \rightarrow \mathbb{R}$. La selección de un subconjunto de características puede estar dado bajo tres consideraciones, según Molina *et al.* (2002):

- Encontrar $D' \subset D$, tal que $J(D')$ es máximo y $|D'| = m$ con $m < n$.
- Encontrar $D' \subseteq D$ con el menor $|D'|$, tal que $J(D') \geq J_o$, con J_o definido como el mínimo valor aceptado para J.
- Encontrar un compromiso entre minimizar $|D'|$ y maximizar $J(D')$ (Caso general).

Un algoritmo de selección de características está conformado, generalmente, por un método de búsqueda, un método para generar sucesores y una medida de evaluación. Los métodos de búsqueda más empleados son los mecanismos de ranqueo de variables, por su simplicidad y escalabilidad, definidos a continuación.

Métodos de selección de características por filtrado

Kohavi *et al.* (1997) plantea que, el ranqueo de variables se categoriza como un método de filtrado, ya que es un paso de preprocesamiento independiente al algoritmo de aprendizaje.

Según Guyon y Elisseeff (2003), sea X un conjunto de ejemplos, y cada ejemplo $\mathbf{x}_i \in X$ $(i = 1, \ldots, N)$ consistiendo de m variables de entrada $\mathbf{x}_{ik}$, con $k = 1, \ldots, m$, el

ranqueo de variables hace uso de una función de puntuación (medida de evaluación) $S(k)$ calculada sobre los valores de $\mathbf{x}_{ik}$, por convención, se supone que una puntuación alta indica que una variable es relevante, y ordena las variables en orden descendente respecto a $S(k)$.

Una de las funciones de puntuación usualmente empleada es la entropía de Shannon (Shannon y Weaver, 1949). En general, la entropía de una variable $\mathbf{x}_k$ con valores v_k, cada uno con probabilidad $p(v_k)$, está definida como:

$$H(k) = -\sum_{k} p(v_k) log(p(v_k)). \tag{23}$$

Una crítica común en el ranqueo de variables es que se puede obtener un subconjunto de características redundantes, es decir, las variables pueden presentar dependencias entre ellas (Guyon y Elisseeff, 2003). Una noción de redundancia en las variables es la correlación entre estas, la cual puede ser medida sobre el conjunto de datos X usando el coeficiente de correlación de Pearson, definido como:

$$\rho_P(j,k) = \frac{cov(j,k)}{\sqrt{var(j)var(k)}}, \tag{24}$$

donde $cov(j,k)$ es la covarianza entre las variables j y k, y $var(j,k)$ su varianza. Así mismo, se puede usar el coeficiente de correlación de Spearman, definido como:

$$\rho_S(j,k) = 1 - 6\frac{\sum_i^{\eta} \delta_i^2}{\eta^3 - \eta}, \tag{25}$$

donde δ_i es la diferencia de rango entre las variables j y k, y η el número de rangos.

El Algoritmo 2 describe el procedimiento para seleccionar características basado en un método de ranqueo y apoyado por un método de eliminación de variables redundantes por correlación. La función *SORT_BY_H* devuelve un ordenamiento descendente de las variables basado en su valor de entropía. La función *GET_LESS_RANK* devuelve la variable que aparece en segundo lugar, entre las que recibe como argumento, en el ranqueo por entropía.

Algoritmo 2: FS-ENTROPY-CORRELATION

Input: a dataset X with features D, where $|D| = n$; a value of $\alpha \in [0, 1]$
Output: $D' \subset D$,where$|D'| = m$ and $m < n$

```
D' ← ∅
foreach feature d_i in D do
    calculate H(X_{d_i}), Equation 23
end
rank ← SORT_BY_H(D)
foreach d_i, d_j ∈ D, with i ≠ j do
    ccp ← ρ_P(X_{d_i}, X_{d_j}), Equation 24
    ccs ← ρ_S(X_{d_i}, X_{d_j}), Equation 25
    if ccp > α or ccs > α then
        less ← GET_LESS_RANK(d_i, d_j, rank)
        (D' ∪ {d_i, d_j}) \ {less}
        D \ {less}
    end
    else
        D' ∪ {d_i, d_j}
    end
end
return D'
```

Debido a que los métodos de filtrado son independientes del algoritmo de aprendizaje, no se puede observar el rendimiento que tendrá el algoritmo con el subconjunto seleccionado. Existe otro método conocido como selección de características por envoltura, el cual permite evaluar el subconjunto a seleccionar dependiendo del desempeño de un algoritmo de aprendizaje. En la siguiente sección se describe este método.

Métodos de selección de características por envoltura

Un enfoque de selección de características por envoltura, utiliza un algoritmo de aprendizaje como una caja negra, en otras palabras, el algoritmo de aprendizaje es solamente usado como una medida de evaluación del subconjunto seleccionado (Kohavi *et al.*, 1997). Un método de búsqueda usualmente empleado para la selección del subconjunto a evaluar es un algoritmo genético (GA, por sus siglas en inglés).

Un GA es un algoritmo de búsqueda heurístico basado en el concepto de selección natural y genética. La idea de este algoritmo es imitar el proceso biológico de supervivencia del más apto, para evolucionar y encontrar una solución óptima para algún problema utilizando operadores de cruzamiento y mutación (Mitchell, 1998).

El proceso general para encontrar un subconjunto de características con un GA se resume a continuación, además, se refiere al lector al trabajo de Beltran *et al.* (2020) donde se propone un método específico para seleccionar descriptores para la clasificación de AMPs basado en un GA:

1. Inicializar una población P de tamaño r de individuos representados como cromosomas. Regularmente vectores binarios.
2. Usar un algoritmo de aprendizaje J para evaluar cada individuo en P.
3. Seleccionar $P' \subset P$
4. Aplicar un operador de cruzamiento sobre los individuos de P' con probabilidad p_c para obtener un conjunto C de individuos.
5. Aplicar un operador de mutación a los individuos de C con probabilidad p_m.
6. Usar J para evaluar los individuos en C.
7. Seleccionar los individuos a sobrevivir para la siguiente generación de los conjuntos P y C.
8. Repetir desde el paso 3 hasta que se cumpla una condición de parada.
9. Devolver el individuo para el que J sea máximo.

2.2.3. Calidad en los conjuntos de datos para modelación QSAR

Voigt *et al.* (2001) propusieron métricas que permiten analizar bases de datos biológicas, estás métricas pueden emplearse para medir la calidad de un conjunto que se usará en una modelación QSAR (Guerrero-Vázquez, 2019), y se definen a continuación.

Diversidad. Dada una función de similitud, la diversidad de un conjunto está dada como la razón del número de conglomerados que se forman al agrupar los elementos del conjunto con una similitud de al menos α, dividido entre la cardinalidad del conjunto. Esta métrica indica la cantidad de información redundante que contiene un conjunto de datos.

Parentesco. Dada una función de similitud, el parentesco de un conjunto S_1 respecto a un conjunto S_2 se define como la razón del número de elementos en S_1 que

tienen una similitud de, al menos, α con, por lo menos, un elemento en S_2, dividido entre la cardinalidad de S_1. El parentesco de dos conjuntos indica el grado de separabilidad de estos, entre mayor sea el valor en esta métrica, más difícil será separar casos positivos de casos negativos a nivel de secuencia.

Una función de similitud entre dos secuencias s_1 y s_2 puede definirse en términos de un alineamiento global, como se muestra en la siguiente expresión:

$$Similitud(s_1, s_2) = \frac{M(align(s_1, s_2))}{L(align(s_1, s_2))}, \tag{26}$$

donde *align* es una función que calcula el alineamiento global entre las secuencias s_1 y s_2 basado en el algoritmo Needleman-Wunsch (Needleman y Wunsch, 1970) y la matriz de puntuación Blosum62 (Henikoff y Henikoff, 1992). M es una función que calcula el número de coincidencias en el alineamiento, y L es una función que calcula la longitud del alineamiento.

Capítulo 3. Metodología

El presente capítulo expone la metodología seguida en esta investigación. Primero, se describe la construcción del conjunto de datos. Después, se detalla el procedimiento para seleccionar los conjuntos de entrenamiento y prueba. Posteriormente, se mencionan las representaciones numéricas utilizadas para codificar los conjuntos de datos. De igual manera, se presentan los algoritmos de aprendizaje estudiados. Por otra parte, se especifican los métodos de evaluación empleados para comparar el rendimiento de los algoritmos. Por último, se describen los péptidos obtenidos para predecir su potencial actividad antimicrobiana.

3.1. Construcción del conjunto de datos

Para la construcción de un conjunto de datos de AMPs, se obtuvieron secuencias de la base de datos StarPepDB (Aguilera-Mendoza *et al.*, 2019) con la herramienta StarPep toolbox v0.8.4. En este paso se obtuvieron un total de 18930 AMPs, de los cuales 14376 presentan actividad antibacteriana, 6194 antifúngica, 650 antiparasitaria y 4653 antiviral (ver archivo *antimicrobial.fasta*).

Depués, como principalmente los péptidos están compuestos por los 20 aminoácidos estándar (Jiménez y Merchant, 2003) y que el 98.8 % de las secuencias en StarPepDB se encuentran en el intervalo de 5 a 100 aminoácidos de longitud, se eliminaron todas aquellas secuencias que tuvieran una longitud menor a cinco aminoácidos y las que tuvieran en su composición aminoácidos no naturales, tales como: B, J, O, U, X y Z; reduciendo el conjunto de AMPs a 17335 secuencias, de las cuales 13249 presentan actividad antibacteriana, 5764 antifúngica, 541 antiparasitaria y 4281 antiviral (ver archivo *antimicrobial_wo_unaa.fasta*).

Por último, dado que un AMP puede tener asociada más de una función biológica, se descartaron todos aquellos para los cuales se ha validado más de una actividad de las cuatro analizadas. La Figura 6 muestra un diagrama de Venn en el que se puede apreciar el número de secuencias con más de una actividad. Al final, el conjunto de AMPs quedó formado por 12345 secuencias, de las cuales 8278 presentan actividad antibacteriana, 993 antifúngica, 130 antiparasitaria y 2944 antiviral (ver archivo *antimicrobial_final.fasta*).

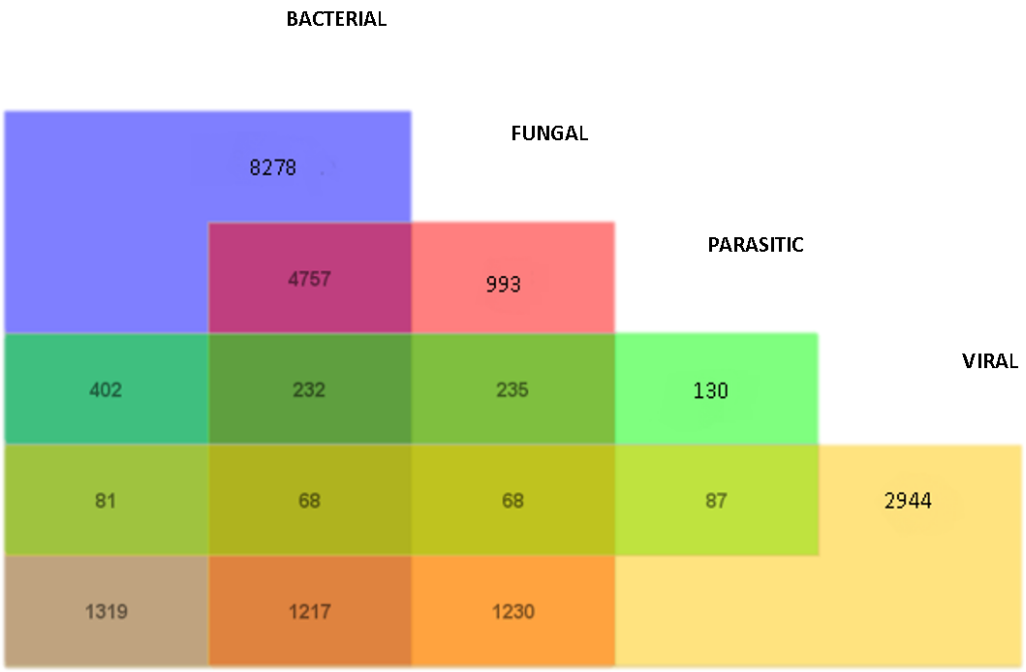

Figura 6. Intersección entre las actividades biológicas estudiadas según las secuencias obtenidas. Figura generada con el software Dover Analizer versión 0.1.2 (Aguilera-Mendoza et al., 2015).

Debido a que se abordó el problema multiclase con un enfoque de descomposición, se tuvieron en total cinco problemas de clasificación binarios, el primero, identificar AMPs, y, después, cuatro problemas binarios para reconocer las actividades antibacteriana, antifúngica, antiparasitaria y antiviral.

Para cada actividad, de las mencionadas anteriormente, se tomó del conjunto de datos de AMPs las secuencias para las cuales se validó cada una de éstas. Los conjuntos finales quedaron constituídos de la siguiente manera: conjunto antibacteriano 8278 secuencias (ver archivo *antibacterial.fasta*), conjunto antifúngico 993 secuencias (ver archivo *antifungal.fasta*), conjunto antiparasitario 130 secuencias (ver archivo *antiparasitic.fasta*) y conjunto antiviral 2944 secuencias (ver archivo *antiviral.fasta*).

A diferencia de los métodos reportados en la literatura, donde usan parte o el total del complemento de cada categoría, al descomponer un problema multiclase, como su conjunto opuesto (negativo) (Lorena *et al.*, 2008; Aly, 2005); aquí, bajo la consideración de continuidad SAR (la cual estalece que cambios graduales en la estructura de una molécula genera cambios graduales en su función) (Muratov *et al.*, 2020), se crearon cinco conjuntos de datos con secuencias potencialmente sin las actividades

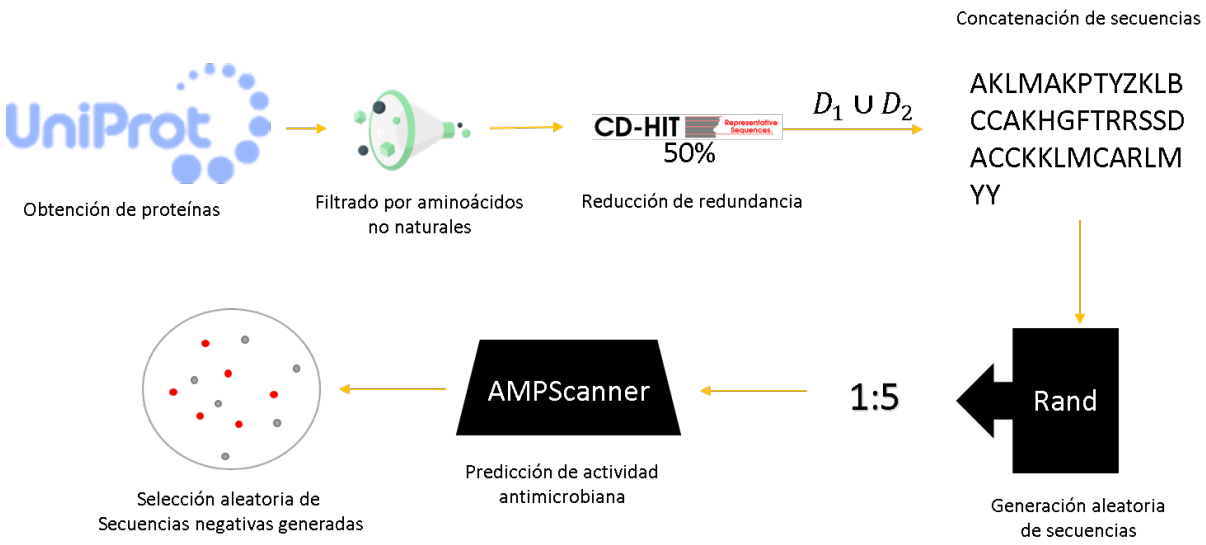

Figura 7. Metodología seguida para generar conjuntos negativos. Diagrama elaborado usando el software Microsoft Power Point versión 2016.

analizadas.

Lo anterior, se debe a que la ausencia de la anotación de la actividad de un péptido a un objetivo particular no garantiza la ausencia de actividad en el péptido. Además, no existen actualmente repositorios públicos de péptidos que demuestren experimentalmente que carecen de las actividades estudiadas. La construcción de estos conjuntos se hizo bajo un enfoque similar a los trabajos previos de Xiao *et al.* (2013) y Gabere y Noble (2017), dicha construcción se describe a continuación:

1. Se descargaron dos conjuntos de proteínas de la base de datos Uniprot versión 2019_08 (UniProt, 2019) utilizando las siguientes cadenas de búsqueda: "*(Golgi OR cytoplasm OR 'endoplasmic reticulum' OR mitochondria) AND NOT antimicrobial, length: [5 TO 100]*" y "*NOT antimicrobial AND reviewed: YES, length: [5 TO 100]*". Obteniendo 506821 y 54225 proteínas, respectivamente, para cada una. La primer cadena de búsqueda se centra en proteínas intracelulares, suponiendo que estas no son dañinas para la célula. La segunda cadena se centra en aquellas proteínas anotadas manualmente como no antimicrobianas.

2. Para tener consistencia con el conjunto positvo y con las mismas razones expuestas anteriormente, se eliminaron todas aquellas secuencias que tuvieran aminoácidos no naturales.

3. Con el programa CD-HIT (Huang *et al.*, 2010) se filtraron, de forma independiente para cada conjunto obtenido de Uniprot, aquellas proteínas que compartieran, al menos, 50 % de identidad de secuencia, manteniendo las representativas. Posteriormente se obtuvo la unión entre los conjuntos reducidos resultantes.

4. Con el conjunto obtenido del paso anterior se generó una súper cadena, la cual es la concatenación de todas las secuencias (cadenas) del conjunto.

5. De forma aleatoria, independientemente para cada actividad, por cada secuencia en el conjunto positivo, se generaron 5 secuencias de la misma longitud obtenidas de una posición aleatoria en la súper cadena. Con esto se crearon cuatro conjuntos negativos independientes.

6. Cada uno de los conjuntos negativos construidos en el paso anterior, se filtraron usando el servidor AMP Scanner (Veltri *et al.*, 2018), eliminando todas las secuencias predichas como AMPs. Con este paso se trató de maximizar el número de secuencias potencialmente sin las actividades deseadas, considerando que las clases estudiadas comparten actividad antimicrobiana.

7. Por último, de cada conjunto negativo filtrado en el paso anterior, se seleccionaron, de forma aleatoria, la misma cantidad de secuencias respecto a su contraparte, el conjunto positivo (ver archivos *nonantibacterial.fasta*, *nonantifungal.fasta*, *nonantiparasitic.fasta* y *nonantiviral.fasta*).

La Figura 7 resume de forma gráfica la metodología descrita.

La unión de estos cuatro conjuntos negativos constituyó el conjunto negativo para el problema de discriminar AMPs de non-AMPs (ver archivo *nonantimicrobial.fasta*).

3.1.1. Partición de los conjuntos de datos en subconjuntos de entrenamiento y prueba

En este trabajo la separación, a diferencia de lo que se reporta en la literatura donde la partición es puramente aleatoria, se basó en el espacio químico de los péptidos; la Figura 8 ilustra el proceso seguido. Este procedimiento se repitió de forma independiente para los conjuntos antibacteriano, antifúngico, antiparasitario y antiviral.

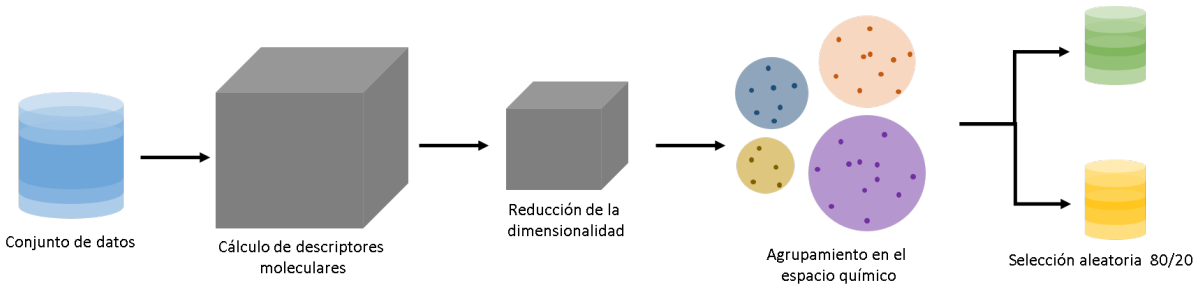

Figura 8. Metodología seguida para seleccionar conjuntos de entrenamiento y prueba. Diagrama elaborado usando el software Microsoft Power Point versión 2016.

Primero, se proyectaron todas las secuencias a su espacio químico con la herramienta de extracción de características disponible en StarPep toolbox v0.8.4 (Aguilera-Mendoza *et al.*, 2019), la cual calcula más de 4000 descriptores moleculares, con sus parámetros por omisión.

Para reducir la dimensión del espacio generado al calcular los descriptores moleculares, se usaron las herramientas de selección de características también disponibles en el mismo software, las cuales tienen un funcionamiento similar al Algoritmo *FS_ENTROPY_CORRELATION* (ver Algoritmo 2), pero además aplica un algoritmo de optimización de subconjuntos. Se usaron los parámetros por omisión del software, únicamente se ajustó el coeficiente de correlación de Spearman con un umbral de 0.95.

Sobre los descriptores moleculares generados en cada conjunto de datos, sin separar las secuencias positivas de las negativas, se aplicó el algoritmo EMC (ver sección 2.1.2). Para esto, se usó la implementación en la herramienta WEKA versión 3.8.4 (Frank *et al.*, 2016), con el parámetro *maximumNumberOfClusters* a -1, para no definir de manera arbitraria un número de grupos, y el resto de parámetros se usaron con sus valores por omisión. La Tabla 1 resume el número de agrupamientos encontrados en cada conjunto de datos.

Tabla 1. Número de grupos encontrados por el algoritmo EMC.

Actividad	**No. Secuencias**	**No. de grupos**
Antibacteriana	16556	48
Antifúngica	1986	20
Antiparasitaria	260	8
Antiviral	5888	44

Una vez obtenidos los grupos, se tomó el 80 % de secuencias positivas y el 80 % de secuencias negativas de forma aleatoria de cada uno de estos, formando con la

unión de estas porciones el conjunto de entrenamiento. El 20 % restante de cada grupo constituyó el conjunto de prueba.

La unión de los conjuntos de prueba para cada actividad, constituyó el conjunto de prueba para el problema de reconocer AMPs. El conjunto de entrenamiento, para este problema, se formó con la unión de los conjuntos de entrenamiento de cada actividad y, además, se añadieron aquellas secuencias anotadas como antimicrobianas en StarPepDB y que tienen asociadas más de una actividad de las analizadas en este trabajo, así mismo, las que no se habían considerado por tener otras actividades como antibiopelículas, anticáncer, antiinflamatoria, entre otras.

Una vez creados los conjuntos de entrenamiento para cada una de las actividades, se analizaron métricas de calidad sobre estos, las métricas calculadas fueron el parentesco y la diversidad (ver sección 2.2.3). El análisis de estas métricas se llevó a cabo con el software Dover Analyzer v 0.1.2 (Aguilera-Mendoza *et al.*, 2015), empleando las funciones de razón de diversidad y traslape de similitud (equivalente al parentesco); con los umbrales de diversidad de 0.05 a 0.95 y los umbrales de traslape de similitud de 0.3 a 0.9.

3.1.2. Construcción de conjuntos de evaluación externos

Con la intención de evaluar la capacidad de generalización de los modelos de clasificación, se crearon conjuntos externos de validación. El término *externos* hace alusión a que las secuencias de estos conjuntos no estaban incluídas en ninguno de los conjuntos de entrenamiento y prueba. Estos conjuntos están constituidos únicamente por secuencias positivas, es decir, secuencias para las cuales se han validado las actividades estudiadas y se conformaron como se describe a continuación:

- Con los AMPs descartados por tener asociada más de una actividad de las cuatro estudiadas (ver sección 3.1), se formaron tres conjuntos externos. El primero, formado por secuencias con actividad antibacteriana y antifúngica, el cual se llamó bacterial_fungal. El segundo, formado por secuencias con actividad antifúngica y antiviral, llamado fungal_viral. El tercero, formado por secuencias con actividad antiparasitaria y antibacteriana, llamado parasitic_bacterial, contenien-

do un total de 4757, 1230 y 411 secuencias, respectivamente (ver archivos *bacterial_fungal.fasta*, *fungal_viral.fasta* y *parasitic_bacterial.fasta*).

- Se descargaron secuencias de la base de datos BIOPEP-UWM (Minkiewicz *et al.*, 2019), para las cuales se ha validado actividad antibacteriana, antifúngica y antiviral (esta base de datos no contiene secuencias con actividad antiparasitaria). Posteriormente, se filtraron por longitud y composición de aminoácidos no naturales. Por último, se eliminaron aquellas que ya se encontraban en alguno de los conjuntos formados previamente. Los conjuntos resultantes se llamaron biopep_bacterial, biopep_fungal y biopep_viral, según la actividad asociada al conjunto obtenido. Al final se obtuvieron 201 secuencias antibacterianas, 47 antifúngicas y 10 antivirales (ver archivos *biopep_bacterial.fasta*, *biopep_fungal.fasta* y *biopep_viral.fasta*).

Para evaluar los clasificadores de AMPs, se construyeron dos conjuntos externos basados en los descritos previamente. El primero, se formó con la unión de los conjuntos bacterial_fungal, fungal_viral y parasitic_bacterial eliminando las secuencias que se compartían a las agregadas al conjunto de entrenamiento (ver archivo *amp_external.fasta*). El segundo, se creó al unir los conjuntos biopep_bacterial, biopep_fungal y biopep_viral (ver archivo *amp_biopep.fasta*).

Adicionalmente, se obtuvieron dos conjuntos reportados por Gabere y Noble (2017), mismos que fueron filtrados para evitar tener secuencias que ya se encontraban en los conjuntos previamente considerados para entrenamiento o evaluación. Después, se tomó la unión de estos para formar un solo conjunto. Por último, por el desbalance entre las clases positivas y negativas, se seleccionó, de manera aleatoria uniforme, del conjunto negativo la misma cantidad de secuencias que en el conjunto positivo, obteniendo de esta forma un conjunto con 554 secuencias positivas y 554 secuencias negativas (ver archivo *gaberenoble.fasta*).

3.2. Representaciones numéricas de los conjuntos de datos

Se consideró el algoritmo PseAAC (ver sección 2.2.2.1), reportado en múltiples trabajos (Xiao *et al.*, 2013; Lin y Xu, 2016; Meher *et al.*, 2017; Bhadra *et al.*, 2018; Lin

et al., 2019; Chung *et al.*, 2019; Yan *et al.*, 2020) que abordan la clasificación de péptidos antimicrobianos, o funciones atribuidas a estos; para representar los conjuntos de datos como vectores numéricos. Para calcular los descriptores de PseAAC se usó la herramienta independiente del servidor iLearn (Chen *et al.*, 2020), una librería para el lenguaje Python en su versión 3.7, ajustando el valor de $\lambda = 2$ y el valor de $\omega = 0.05$.

Por otra parte, se utilizó el software ProtDCal (ver sección 2.2.2.1) para calcular descriptores moleculares y obtener otras 8 representaciones diferentes de los conjuntos de datos, con diversos operadores de vecindad (OVs) implementados en esta herramienta. La Tabla 2 muestra los parámetros seleccionados para obtener dichas codificaciones.

Tabla 2. Parámetros seleccionados en el software ProtDCal para calcular descriptores moleculares por OV.

OV	Cutoff	Índices	Grupos	OA
None	No aplica		ALA, GLN, LEU, SER,	
Kier Hall	No aplica		ARG, GLU, LYS, THR,	
Estado electrotopológico	No aplica	Gs(U), Gw(U), W(U),	ASN, GLY, MET, TRP,	N1, N2, N3, Ar, P2,
	1	Mw, HP, Pa, Z1, ΔHf,	ASP, HIS, PHE, TYR,	P3, M, G, K, RA, S, V,
	2	Ap, IP, Pb, Z2, Xi,	CYS, ILE, PRO, VAL,	CV, DE, MN, MX, Q1,
Autocorrelación	3	ECI, ISA, Pt, Z3, L1-9	AHR, PCR, ARM,	Q2, Q3, I50, SI, MI,
	4		NPR, BSR, NCR, ALR,	TI
	5		PLR, RTR, UCR, UFR	

Las representaciones numéricas generadas con ProtDCal fueron nombradas según el OV utilizado y el *cutoff* en el caso de autocorrelación, como se muestra en la Tabla 3. Una vez obtenidos los descriptores moleculares de ProtDCal, como la herramienta puede devolver valores nulos (-9999), se sustituyeron los valores nulos por ceros.

Tabla 3. Numenclatura usada para referirse a cada codificación generada con descriptores de ProtDCal y número de descriptores obtenidos con cada configuración de parámetros.

Operador de vecindad	Cutoff	Nomenclatura	No. de descriptores calculados
None	No Aplica	NO	6181
Kier Hall	No Aplica	KH	12835
Estado electrotopológico	No Aplica	ES	12835
	1	AC1	12835
	2	AC2	12835
Autocorrelación	3	AC3	12835
	4	AC4	12835
	5	AC5	12835

Las representaciones basadas en descriptores moleculares fueron propuestas para usar con los algoritmos clásicos de aprendizaje. Para el caso de los algoritmos profundos, se usó una representación como la propuesta por Veltri *et al.* (2018). Primero, las

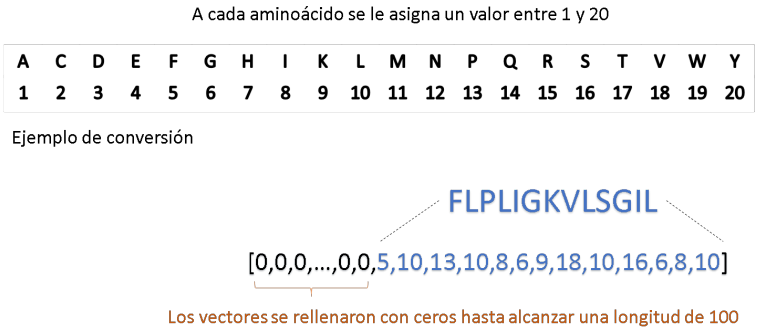

Figura 9. Codificación de secuencias a vectores numéricos para entrenar modelos profundos. Figura adaptada de (Veltri et al., 2018), realizada usando el software Microsoft Power Point versión 2016.

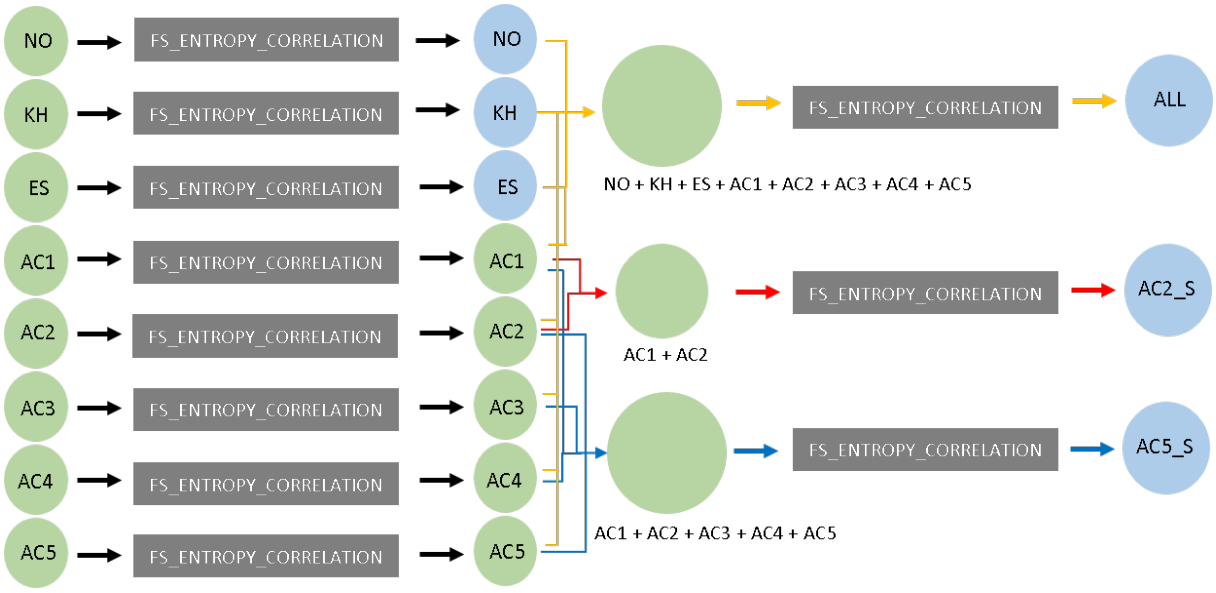

Figura 10. Proceso de selección de características sobre las representaciones basadas en descriptores moleculares de ProtDCal descritas en la sección 3.2. Diagrama elaborado usando el software Microsoft Power Point versión 2016.

secuencias de cada conjunto de datos fueron codificadas a un vector numérico, en el cual cada aminoácido está representado por un valor entero. Por último, los vectores se rellenaron con ceros al inicio hasta conseguir la longitud de la secuencia más larga del conjunto (100 aminoácidos). La Figura 9 muestra esta conversión en forma gráfica.

3.2.1. Selección de características en descriptores moleculares calculados con ProtDCal

Considerando la dimención de las codificaciones generadas con descriptores moleculares de ProtDCal, se llevó a cabo un proceso de selección de características sobre ellas. La Figura 10 ilustra el procedimiento llevado a cabo.

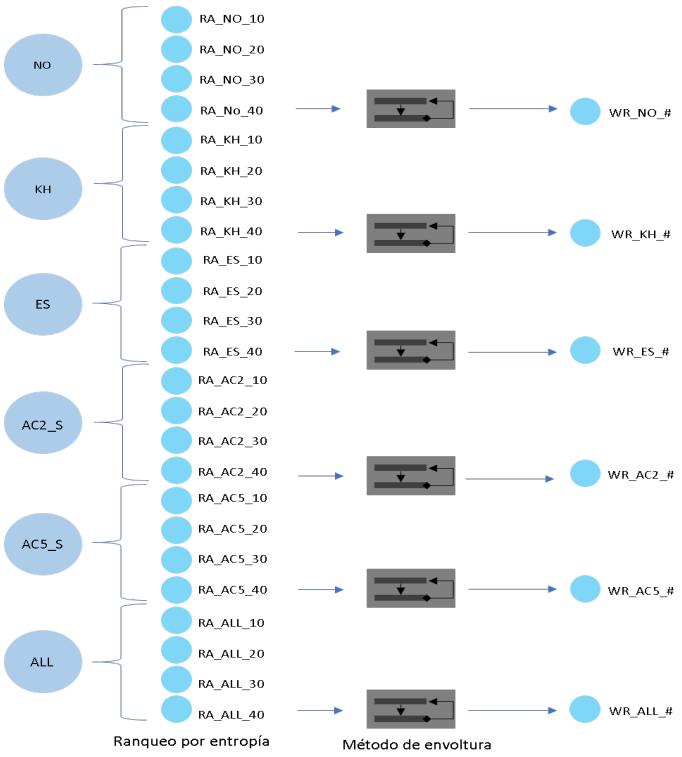

Figura 11. Procedimiento para seleccionar conjuntos finales a usar con algoritmos de aprendizaje. El número de descriptores que se obtuvieron con el método de envoltura es variable para cada conjunto. Diagrama elaborado usando el software Microsoft Power Point versión 2016.

Primero, sobre cada una de las representaciones generadas con ProtDCal, se aplicó el algoritmo *FS_ENTROPY_CORRELATION* (ver Algoritmo 2) ajustando el valor α=0.95. El Anexo A, Tabla 29, muestra el número de descriptores seleccionados en este paso. Una vez realizado el filtro anterior, se unieron las bases reducidas AC1 y AC2 sin sustituir a estas, al conjunto resultante se le aplicó el mismo proceso de selección, para formar así un único conjunto denominado AC2_S.

De forma similar a la anterior, se construyó el conjunto AC5_S, el cual es la unión de las bases reducidas AC1, AC2, AC3, AC4 y AC5, filtrado posteriormente con el algoritmo mencionado. Por último, se unieron las bases de descriptores moleculares NO, KH, ES, AC1, AC2, AC3, AC4 y AC5, sobre el conjunto obtenido, se empleó el mismo algoritmo, obteniendo así un conjunto al que se le denominó ALL.

Al final, se consideraron un total de 6 representaciones basadas en los descriptores moleculares de ProtDCal filtrados, estos fueron los cojuntos NO, KH, ES, AC2_S, AC5_S y ALL. El Anexo A, Tabla 30, muestra el número de descriptores seleccionados para cada codificación.

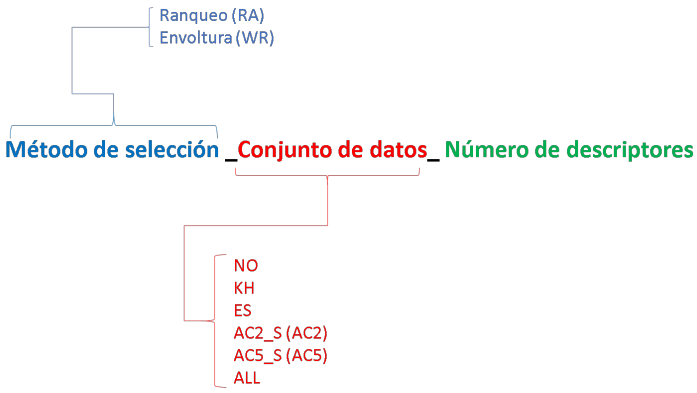

Figura 12. Construcción de la nomenclatura para referirse a cada una de las 30 codificaciones usadas. Figura realizada con el software Microsoft Power Point versión 2016.

Posteriormente, se realizó el proceso mostrado en la Figura 11 para seleccionar subconjuntos de descriptores de cada representación considerada. Tomando en cuenta el ranqueo por entropía obtenido con el algoritmo de selección de características, se conformaron cuatro conjuntos de descriptores moleculares independientes por cada representación, formados por las primeras 10, 20, 30 y 40 características del ranqueo.

Aunado a lo anterior, sobre la representación formada por los 40 descriptores moleculares con mayor entropía, se aplicó una técnica de selección de características de envoltura (ver sección 2.2.2.2) implementada en la herramienta WEKA versión 3.8.4, con el nombre *WrapperSubsetEval*, se seleccionó como método de búsqueda *GeneticSearch* y se ajustó el parámetro *classifier* a *RandomForest*, el resto de parámetros se usaron en su configuración por omisión. Con los descriptores seleccionados tras este procedimiento se formó una nueva codificación, sin sustituir el conjunto original con las 40 características mejor ponderadas.

Con esto, para cada actividad biológica, se tuvieron un total de 30 diferentes representaciones basadas en descriptores moleculares calculados con ProtDCal. La Figura 12 muestra la construcción de la nomenclatura usada para cada codificación.

3.3. Modelos de clasificación basados en algoritmos de aprendizaje clásicos

En este trabajo se empleó el algoritmo RF (ver sección 2.1.1.1) para entrenar, construir modelos de clasificación y predecir las actividades en estudio; se usó la implementación de este en la herramienta WEKA versión 3.8.4, se ajustó el parámetro *numIterations* = 200 (número de árboles), y el resto de configuraciones, profundidad de

los árboles y número de descriptores a seleccionar en cada nodo, con sus valores por omisión.

Se entrenaron 31 modelos de clasificación para cada una las actividades antibacteriana, antifúngica, antiparasitaria y antiviral, para ello, se usaron los conjuntos de entrenamiento respectivos, codificados con las representaciones obtenidas con los descriptores de PseAAC y ProtDCal (ver sección 3.2).

Además, considerando el rendimiento mostrado en un proceso de validación cruzada por los clasificadores construidos, se seleccionaron los cuatro modelos con mejor desempeño, independientemente para cada actividad, y se formó un modelo por ensamble, para el cual, la predicción está dada como el promedio de las probabilidades de pertenecer a la clase positiva o negativa generadas por cada modelo individual.

Aunado a esto, con las representaciones con las cuales se entrenaron los modelos seleccionados para el ensamble; se tomó la unión de estas y sobre el conjunto resultante se llevó a cabo un proceso de selección de características por envoltura (igual al descrito en la sección 3.2.1), el Anexo A, Tabla 31 muestra la cantidad de descriptores obtenidos tras este procedimiento. Con los descriptores seleccionados se entrenó un clasificador con el algoritmo RF; realizando este procedimiento de forma independiente para cada actividad.

Para el caso de la actividad antimicrobiana, se construyeron dos modelos de clasificación basados en el algoritmo RF, el primero, entrenado con descriptores de PseAAC. Para la construcción del segundo, se consideró la unión de los descriptores moleculares usados en los clasificadores elegidos para el ensamble de cada actividad, y sobre el conjunto obtenido se aplicó un proceso de selección de características por envoltura, como el descrito previamente (ver sección 3.2.1). Se tomó el conjunto resultante de este procedimiento como representación del conjunto de datos (representación denominada WR_AMP_131). Los descriptores seleccionados se aprecian en el Anexo A, Tabla 32.

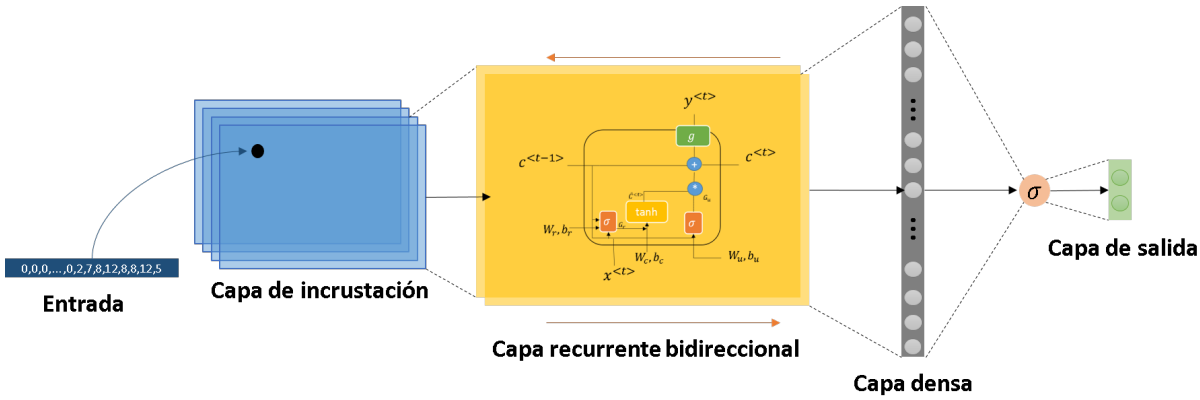

Figura 13. Estructura de la red neuronal propuesta. Cada secuencia se codificó como un vector numérico de longitud 100, estos vectores alimentaron una capa de incrustación la cual mapea estos vectores a un espacio n-dimensional diferente para formar un nuevo vector el cual es la entrada a una capa recurrente bidireccional, la salida de la capa recurrente es pasada a una capa densa y por último la salida de la red es calculada por una función logística. Diagrama elaborado usando el software Microsoft Power Point versión 2016.

3.4. Modelos de clasificación basados en algoritmos de aprendizaje profundo

En este trabajo se emplearon DNNs (ver sección 2.1.1.2) para construir los modelos profundos, sin embargo, cabe mencionar que el término profundo hace referencia a la arquitectura del algoritmo más que al número de capas que el modelo propuesto contiene.

La Figura 13 muestra la arquitectura utilizada para crear los modelos de clasificación, esta contiene cuatro capas, la primera, una capa de incrustación, esta permite crear una representación más compacta del alfabeto (aminoácidos), esto podría ayudar a reducir el tamaño del espacio de búsqueda de secuencias para el diseño de nuevos AMPs (Veltri *et al.*, 2018) en trabajos futuros. La segunda, una capa bidireccional con unidades recurrentes, que permite identificar dependencias en el orden de los aminoácidos de inicio a fin y en sentido opuesto en las secuencias. Posterior a esta, una capa densa que alimenta la capa de salida que predice la actividad en cuestión.

Se usaron unidades GRU y LSTM como unidades recurrentes en la capa bidireccional, esta decisión está apoyada en la capacidad de estas arquitecturas para *reconocer* y *olvidar* patrones importantes en la secuencia (ver sección 2.1.1.3). La implementación de la red propuesta, así como los distintos tipos de unidades recurrentes, se hizo utilizando el framework TensorFlow v2.1.0 (Abadi *et al.*, 2016) y el API de Keras v2.2.4

(Chollet *et al.*, 2015).

Considerando el amplio espacio de hiperpárametros que se necesitan optimizar para mejorar el desempeño de una DNN, se llevó a cabo un proceso de optimización por búsqueda en rejilla usando la herramienta HParams de tensorboard, la Tabla 4 muestra el espacio de parámetros sobre el cual se realizó el procedimiento mencionado. Considerando la complejidad computacional del método de búsqueda utilizado, se definió un espacio de hiperparámetros que se pudiera explorar con la infraestructura disponible.

Tabla 4. Espacio definido para el proceso de búsqueda en rejilla.

Hiperparámetro	**Espacio definido**
Longitud del incrustamiento	[32, 64, 128, 256]
Tipo de Unidad*	[‘GRU’, ‘LSTM’]
No. de Unidades recurrentes	[32, 64, 128, 256]
Tasa dropout	[0.3, 0.5]
No. de Unidades densas	[32, 64, 128, 256]
Optimizador	[’adam’, ’sgd’]

* El tipo de unidad se consideró como un hiperparámetro.

3.5. Evaluación y comparación del desempeño de los modelos de clasificación basados en algoritmos clásicos y profundos

Para medir el rendimiento de los clasificadores se utilizaron las métricas SN, SP, ACC y MCC (ver sección 2.1). Para observar el comportamiento y capacidad de generalización de los modelos de clasificación clásicos y profundos, en lo individual, las métricas referidas anteriormente se calcularon basadas en un proceso de validación cruzada de 10 pliegues, sobre cada conjunto de entrenamiento de los cinco problemas de clasificación binaria estudiados.

Para comparar el rendimiento entre los modelos construidos con algoritmos clásicos de clasificación y los basados en algoritmos profundos, en cada uno de los problemas de clasificación analizados, se realizó un procedimiento de validación por reserva, usando el 80 % del conjunto de entrenamiento para construir el modelo, y el 20 % restante para evaluar. Este proceso se repitió 30 veces modificando las particiones generadas, utilizando las mismas divisiones de los datos tanto en los modelos clásicos como en los profundos. Además de esto, se construyeron clasificadores usando el conjunto de entrenamiento completo, y fueron evaluados utilizando los conjuntos de

prueba y externos (ver sección 3.1.1 y 3.1.2).

3.6. Comparación del rendimiento entre los modelos propuestos en este trabajo y modelos reportados en la literatura

Se usaron los conjuntos de prueba y externos (ver sección 3.1.1 y 3.1.2) propuestos en este trabajo para comparar el rendimiento entre los modelos propuestos en esta investigación y algunos modelos reportados en la literatura, según la disponibilidad de los mismos. Para los clasificadores de actividad antibacteriana y antifúngica, se usó el conjunto externo bacterial_fungal, para actividad antiviral el conjunto fungal_viral, y para actividad antiparasitaria el conjunto parasitic_bacterial. Para la comparación de los clasificadores de AMPs se usaron los conjuntos de prueba y externos propios a este problema. Para llevar a cabo una comparación justa, se eliminaron de los conjuntos de prueba y externos propuestos, las secuencias consideradas en los conjuntos de entrenamiento de los modelos de la literatura, independiente por cada actividad.

Los modelos reportados en los servidores iAMP-2L (Xiao *et al.*, 2013), MLAMP (Lin y Xu, 2016) y AMPfun (Chung *et al.*, 2019), se consideraron en la comparación de los clasificadores propuestos para actividad antimicrobiana, antibacteriana, antifúngica y antiviral. Además, AMPfun se consideró para la clase antiparasitaria.

Los clasificadores de iAMPpred (Meher *et al.*, 2017) y ClassAMP (Joseph *et al.*, 2012), se consideraron en la evaluación de los clasificadores de actividad antibacteriana, antifúngica y antiviral. Del servidor PEPred-Suite (Wei *et al.*, 2019), se usaron los modelos para actividad antibacteriana y antiviral. AMPScanner (Veltri *et al.*, 2018), se empleó en la comparación de las clases antimicrobiana y antibacteriana.

De la misma manera, los modelos reportados en ADAM (Lee *et al.*, 2015) y CAMPR3 (Waghu *et al.*, 2016), se usaron en las comparaciones de actividad antimicrobiana. Los servidores ABP-Finder (Romero-Molina *et al.*, 2019) y AntiFP (Agrawal *et al.*, 2018) se usaron en las comparaciones de actividad antibacteriana y antifúngica, respectivamente.

3.6.1. Comparación basada en un enfoque jerárquico

Algunos servidores reportados en la literatura, contienen modelos de clasificación que siguen una estructura jerárquica (Xiao *et al.*, 2013; Lin y Xu, 2016; Chung *et al.*, 2019), es decir, tienen dos niveles de clasificación. En el primer nivel, estos modelos identifican secuencias con actividad antimicrobiana. Posteriormente, las secuencias predichas como AMPs, son pasadas por clasificadores de actividades asociadas a estos, tales como; antibacteriana, antifúngica, antiparasitaria, antiviral, anticáncer, entre otras.

Debido a este enfoque jerárquico, del primer clasificador, se generan falsos negativos, los cuales se acarrean a los cálculos de las métricas de los clasificadores de actividades específicas. Por lo tanto, las métricas de evaluación para las comparaciones con estos servidores se calcularon de dos formas. La primera, sin considerar el enfoque jerárquico, se calcularon las métricas de evaluación acarreando los resultados del primer nivel de clasificación al segundo. La segunda, tomando en cuenta el enfoque jerárquico, las métricas para los modelos antibacterial, antifúngico, antiparasitario y antiviral se calcularon considerando solo las secuencias que se predijeron como AMPs en los modelos jerárquicos.

3.7. Ejemplo de aplicación. Predicción de actividades antimicrobianas en un metagenoma de esponjas marinas del Parque Nacional Cabo Pulmo

En los últimos años ha aumentado la evidencia de que bacterias marinas sintetizan compuestos valiosos para el descubrimiento de fármacos (Gulder y Moore, 2009; Rahman *et al.*, 2010; Esteves *et al.*, 2013). Las esponjas marinas se han considerado como los organismos marinos con mayor producción de metabolitos secundarios, asociando a muchos de ellos un origen bacteriano (Teta *et al.*, 2010). Las actinobacterias son un grupo presente en las esponjas marinas (Lago-Lestón *et al.*, 2013); miembros de este grupo han mostrado gran potencial para el desarrollo de fármacos (Bull y Stach, 2007; Dharmaraj, 2010). Lo anterior motiva a analizar el potencial biotecnológico que estos meta-proteomas podrían tener.

Con la intención de proponer nuevas secuencias con potencial uso terapeútico, se analizaron péptidos predichos previamente como AMPs en el proyecto CICESE denomi-

nado: "*Exploración de la biodiversidad y del potencial biotecnológico de las comunidades microbianas asociadas a esponjas marinas en Cabo Pulmo, mediante tecnologías de vanguardia*". Estos péptidos se generaron de un metagenoma de especies de esponjas marinas del Parque Nacional Cabo Pulmo. Se puede encontrar una relación de estos péptidos en el Anexo B, Tabla 34.

Cada una de las secuencias recabadas del proyecto mecionado, se codificaron a las diferentes representaciones numéricas usadas en este trabajo y se predijo su actividad usando los clasificadores con mejor rendimiento para cada una de las actividades estudiadas, incluyendo la actividad antimicrobiana.

Capítulo 4. Resultados

En este capítulo se describen los experimentos realizados para cumplir con el objetivo de evaluar el desempeño de los modelos de aprendizaje propuestos para el problema de clasificación multiclase de péptidos antimicrobianos. Así mismo, se muestran los resultados obtenidos de los experimentos elaborados y la comparación de dichos resultados contra los obtenidos con métodos del estado del arte.

4.1. Construcción del conjunto de datos

Siguiendo la metodología presentada en la sección 3.1, se construyeron cinco conjuntos de datos: el conjunto antimicrobiano, antibacteriano, antifúngico, antiparasitario y antiviral, constituidos por 17155/12331, 8278/8278, 993/993, 130/130 y 2944/2944 secuencias positivas/negativas, respectivamente.

Posteriormente, aplicando la metodología descrita en la sección 3.1.1, se dividieron los conjuntos de datos en subconjuntos de entrenamiento y prueba, la Tabla 5 muestra el número de secuencias que conformaron cada uno de estos subconjuntos.

Tabla 5. Número de secuencias en los conjuntos de entrenamiento y prueba para cada una de las actividades estudiadas.

Conjunto	Entrenamiento		Prueba	
	Casos Positivos	Casos Negativos	Casos Positivos	Casos Negativos
Antimicrobiano	14951	9767	2564	2564
Antibacteriano	6583	6583	1695	1695
Antifúngico	778	778	215	215
Antiparasitario	99	99	31	31
Antiviral	2321	2321	623	623

Una vez divididos los conjuntos de datos, sobre los conjuntos de entrenamiento, como se menciona en la sección 3.1.1, se calculó el parentesco y la diversidad (ver sección 2.2.3) para analizar la calidad de los conjuntos. De la misma manera, estas métricas se evaluaron en los conjuntos de entrenamiento disponibles de los modelos reportados en la literatura (ver sección 3.6) para comparar la calidad de los conjuntos propuestos respecto a estos últimos. Las figuras 14 a la 18 muestran los resultados al medir el parentesco, mientras que las figuras 19 a la 23 resumen los resultados al calcular la diversidad de los ejemplos positivos y negativos, por cada una de las actividades estudiadas bajo distintos umbrales de similitud.

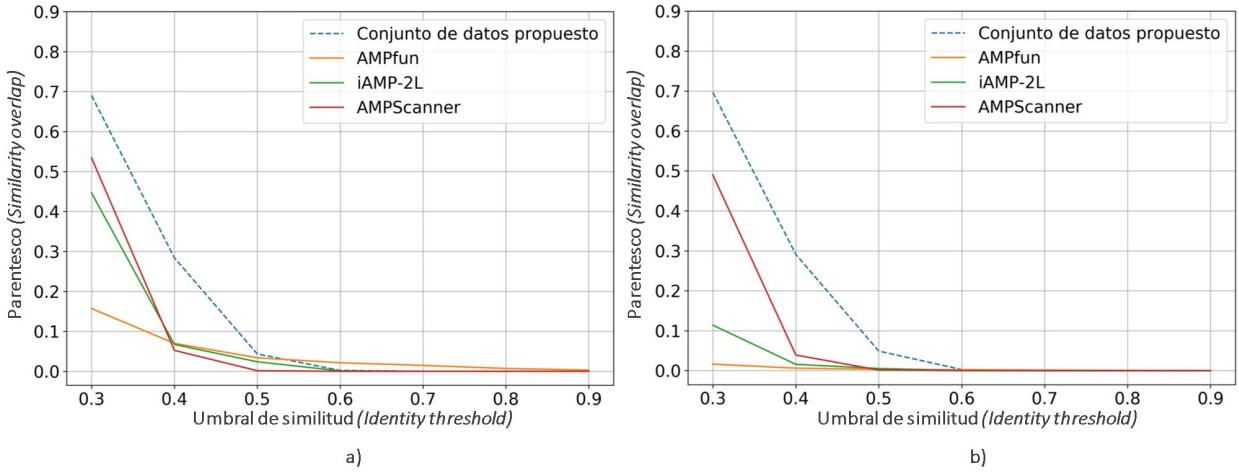

Figura 14. Parentesco entre secuencias positivas y negativas del conjunto de entrenamiento para actividad antimicrobiana. a) Parentesco del conjunto positivo hacia el conjunto negativo. b) Parentesco del conjunto negativo hacia el conjunto positivo.

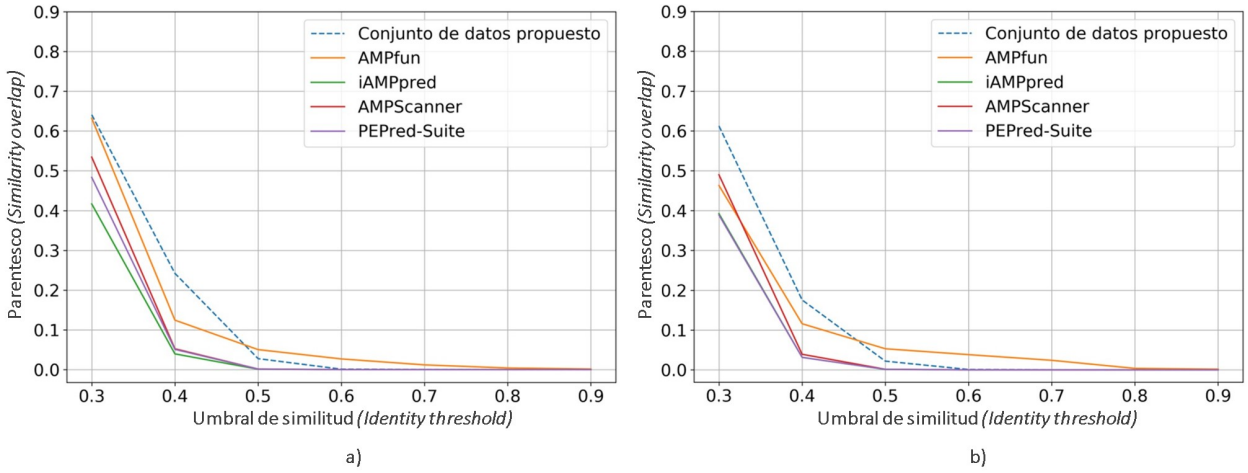

Figura 15. Parentesco entre secuencias positivas y negativas del conjunto de entrenamiento para actividad antibacteriana. a) Parentesco del conjunto positivo hacia el conjunto negativo. b) Parentesco del conjunto negativo hacia el conjunto positivo.

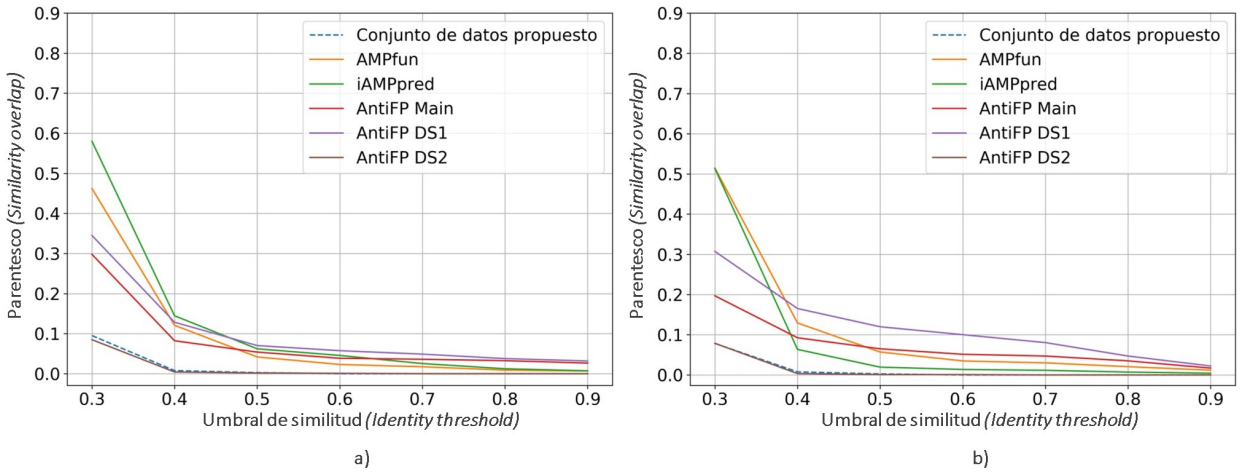

Figura 16. Parentesco entre secuencias positivas y negativas del conjunto de entrenamiento para actividad antifúngica. a) Parentesco del conjunto positivo hacia el conjunto negativo. b) Parentesco del conjunto negativo hacia el conjunto positivo.

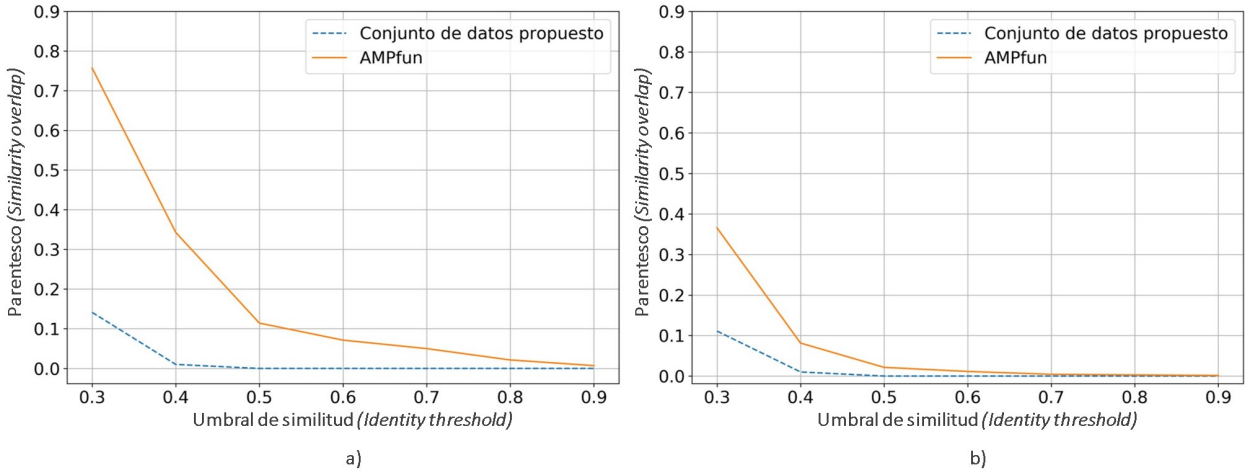

Figura 17. Parentesco entre secuencias positivas y negativas del conjunto de entrenamiento para actividad antiparasitaria. a) Parentesco del conjunto positivo hacia el conjunto negativo. b) Parentesco del conjunto negativo hacia el conjunto positivo.

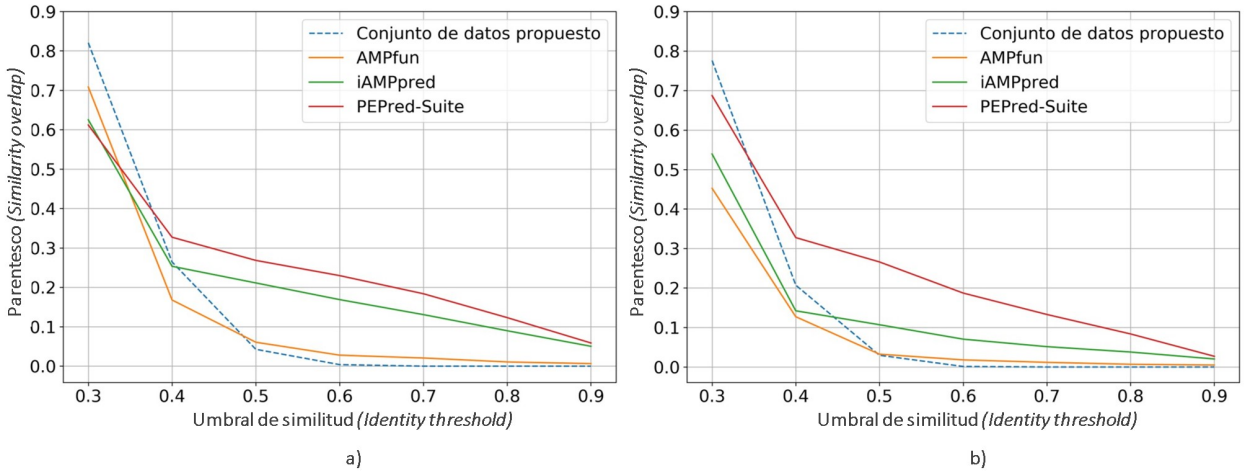

Figura 18. Parentesco entre secuencias positivas y negativas del conjunto de entrenamiento para actividad antiviral. a) Parentesco del conjunto positivo hacia el conjunto negativo. b) Parentesco del conjunto negativo hacia el conjunto positivo.

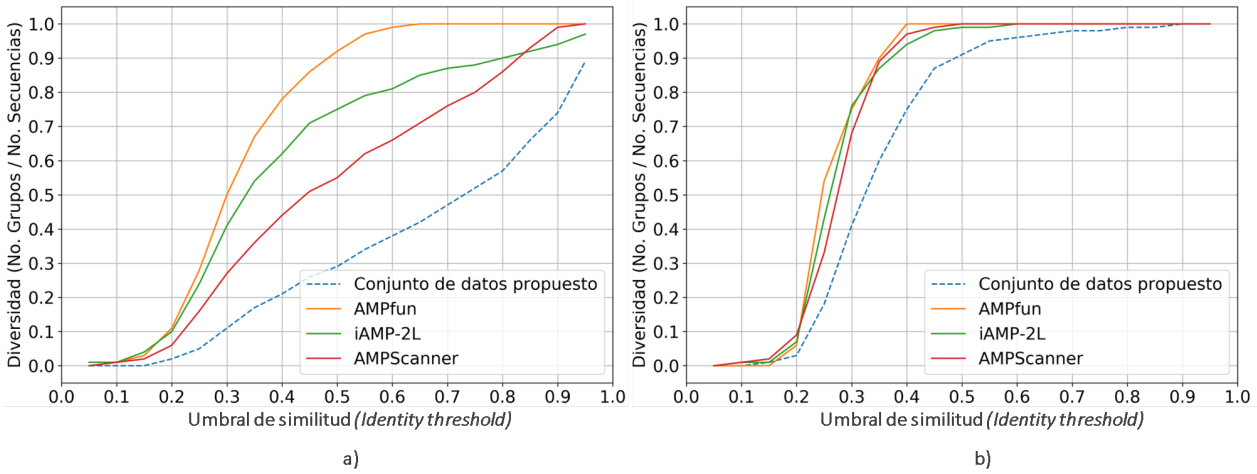

Figura 19. Diversidad del conjunto de datos antimicrobiano. a) Diversidad del conjunto positivo. b) Diversidad del conjunto negativo.

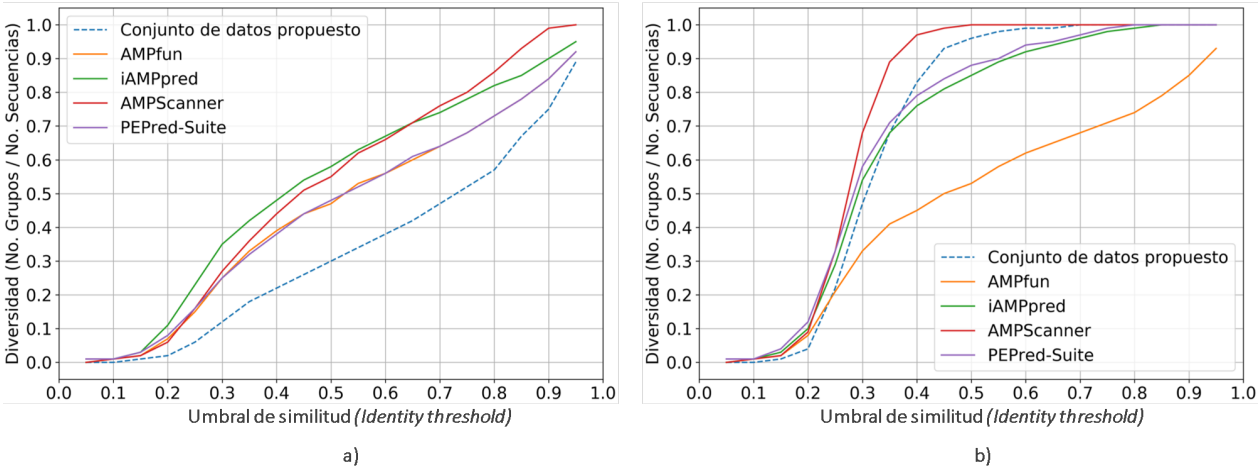

Figura 20. Diversidad del conjunto de datos antibacteriano. a) Diversidad del conjunto positivo. b) Diversidad del conjunto negativo.

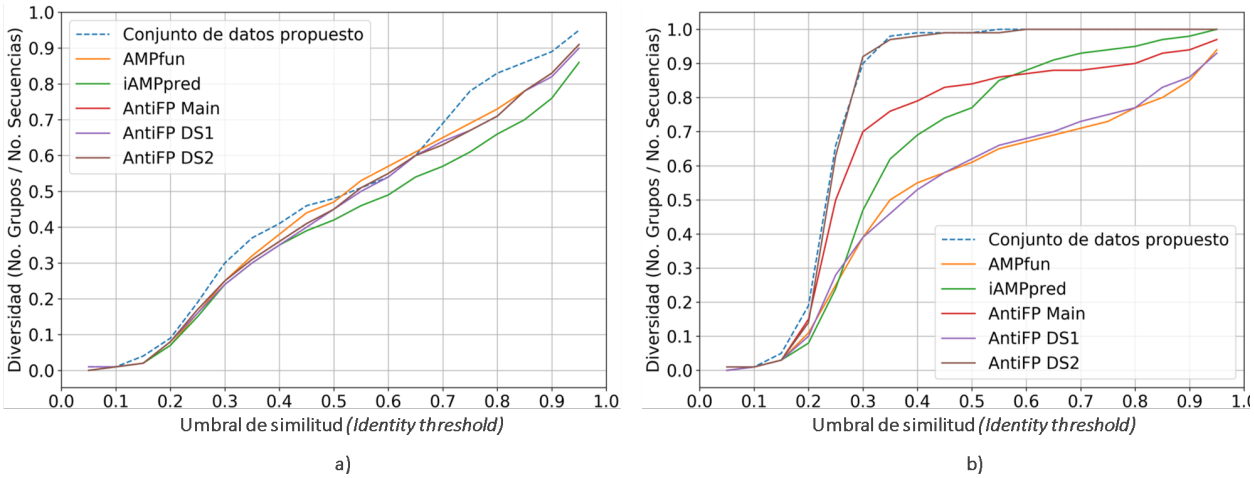

Figura 21. Diversidad del conjunto de datos antifúngico. a) Diversidad del conjunto positivo. b) Diversidad del conjunto negativo.

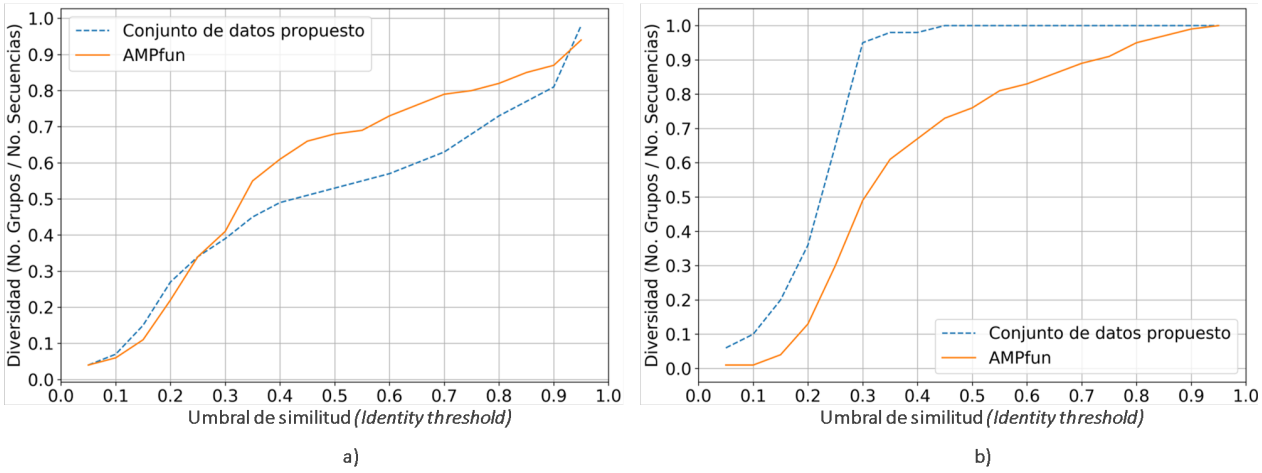

Figura 22. Diversidad del conjunto de datos antiparasitario. a) Diversidad del conjunto positivo. b) Diversidad del conjunto negativo.

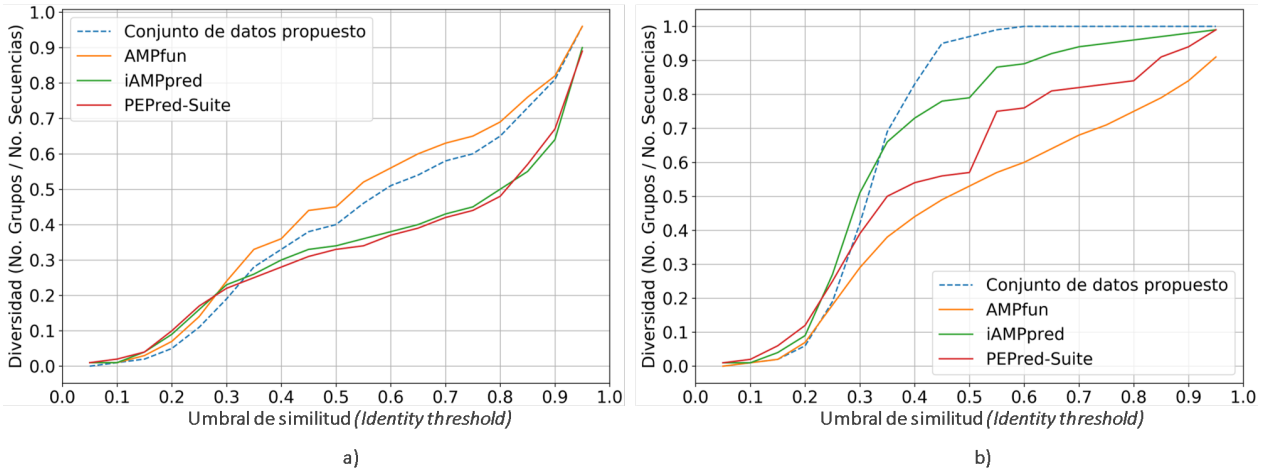

Figura 23. Diversidad del conjunto de datos antiviral. a) Diversidad del conjunto positivo. b) Diversidad del conjunto negativo.

Se observó que los conjuntos antimicrobiano y antibacteriano (figuras 14 y 15), propuestos en este trabajo, tienen mayor parentesco entre casos positivos y negativos (en ambas direcciones) bajo umbrales de similitud del 0.3 al 0.6, comparados con los conjuntos propuestos en la literatura para dichas actividades, a excepción del conjunto del servidor AMPfun. Lo anterior se traduce a conjuntos con mayor complejidad para separarse a nivel de secuencia. Empero, se encuentran entre los conjuntos con menor diversidad, lo que indica que contienen un mayor número de secuencias redundantes (figuras 19 y 20).

Un caso similar a los anteriores se observa en el conjunto propuesto para la clase antiviral (Figura 18), el parentesco de este conjunto es superior a lo alcanzado por sus equivalentes de la literatura a un umbral de similitud de 0.3, con un valor comparable a los otros conjuntos bajo umbrales de 0.4 y 0.5, por ello, también se considera un conjunto difícil de separar a nivel de secuencia. En contraste a lo anterior, los conjuntos antifúngico y antiparasitario (figuras 16 y 17) construídos en esta investigación, presentan un parentesco bajo (10 % en promedio) a un umbral de similitud de 0.3, lo que podría significar que son conjuntos fácilmente separables a nivel de secuencia.

La diversidad de los tres conjuntos mencionados en el párrafo anterior es comparable a la diversidad mostrada por los conjuntos de la literatura para las clases antiviral, antifúngica y antiparasitaria en los casos positivos, y superior en los casos negativos (figuras 21, 22 y 23). Es importante mencionar que a diferencia de lo que se reporta en la literatura, en este trabajo no se usó una técnica para eliminar redundancia basada en secuencia, por el contrario se disminuyó la redundancia en el espacio químico durante la partición en conjuntos de entrenamiento y prueba.

4.2. Desempeño de los modelos clásicos de clasificación

Empleando cada una de las representaciones generadas con descriptores moleculares de ProtDCal (ver sección 3.2), se construyeron clasificadores usando el algoritmo RF siguiendo la metodología descrita en la sección 3.3. Las tablas 6 y 7, detallan los valores obtenidos en las métricas de evaluación bajo un proceso de validación cruzada de 10 pliegues, sobre el conjunto de entrenamiento, para cada clasificador construido por cada actividad. Adicionalmente, el Anexo D, Tabla 36, contiene los valores del error cuadrático medio obtenido por cada clasificador.

Tabla 6. Métricas de un proceso de validación cruzada sobre el conjunto de entrenamiento para cada uno de los clasificadores construidos con las representaciones de ProtDCal por actividad.

Conjunto de entrenamiento	Modelo[a]	SN	SP	ACC	MCC
Antibacteriano	RA_NO_10	0.858	0.909	0.884	0.769
	RA_NO_20	0.863	0.914	0.889	0.778
	RA_NO_30	0.876	0.923	0.899	0.801
	RA_NO_40	0.879	0.930	0.905	0.810
	WR_NO_25*	0.878	**0.937**	**0.907**	**0.816**
	RA_ES_10	0.839	0.878	0.858	0.718
	RA_ES_20	0.857	0.903	0.880	0.761
	RA_ES_30	0.868	0.917	0.893	0.786
	RA_ES_40	**0.881**	0.925	0.903	0.806
	WR_ES_29*	0.878	0.924	0.901	0.803
	RA_KH_10	0.817	0.844	0.830	0.661
	RA_KH_20	0.860	0.911	0.885	0.771
	RA_KH_30	0.870	0.911	0.890	0.781
	RA_KH_40	0.867	0.912	0.889	0.780
	WR_KH_27	0.870	0.914	0.892	0.785
	RA_AC2_10	0.823	0.866	0.845	0.690
	RA_AC2_20	0.851	0.912	0.881	0.764
	RA_AC2_30	0.870	0.926	0.898	0.796
	RA_AC2_40	0.872	0.927	0.900	0.800
	WR_AC2_30*	0.871	0.928	0.899	0.800
	RA_AC5_10	0.817	0.837	0.827	0.654
	RA_AC5_20	0.844	0.896	0.870	0.742
	RA_AC5_30	0.843	0.898	0.871	0.743
	RA_AC5_40	0.849	0.905	0.877	0.754
	WR_AC5_23	0.856	0.906	0.881	0.763
	RA_ALL_10	0.849	0.886	0.868	0.736
	RA_ALL_20	0.863	0.916	0.889	0.780
	RA_ALL_30	0.871	**0.937**	0.904	0.809
	RA_ALL_40	0.869	0.938	0.903	0.809
	WR_ALL_24*	0.875	0.936	0.906	0.813
Antifúngico	RA_NO_10	0.911	0.940	0.925	0.851
	RA_NO_20	0.918	0.954	0.936	0.872
	RA_NO_30	0.918	0.968	0.940	0.881
	RA_NO_40	0.911	0.960	0.936	0.873
	WR_NO_21*	0.923	**0.969**	**0.946**	**0.893**
	RA_ES_10	0.919	0.936	0.927	0.855
	RA_ES_20	0.920	0.950	0.935	0.871
	RA_ES_30	0.915	0.955	0.935	0.871
	RA_ES_40	0.920	0.964	0.942	0.884
	WR_ES_25	**0.925**	0.961	0.943	0.887
	RA_KH_10	0.910	0.943	0.927	0.854
	RA_KH_20	0.906	0.941	0.924	0.848
	RA_KH_30	0.907	0.949	0.928	0.857
	RA_KH_40	0.907	0.952	0.930	0.861
	WR_KH_25	0.914	0.956	0.935	0.871
	RA_AC2_10	0.875	0.922	0.898	0.798
	RA_AC2_20	0.900	0.943	0.922	0.844
	RA_AC2_30	0.906	0.941	0.924	0.848
	RA_AC2_40	0.913	0.952	0.933	0.866
	WR_AC2_23*	0.905	0.946	0.925	0.852
	RA_AC5_10*	0.895	0.920	0.907	0.815
	RA_AC5_20	0.901	0.942	0.922	0.844

Continúa en la siguiente página

Tabla 6 – Continuación de la página anterior

Conjunto de entrenamiento	Modelo[a]	SN	SP	ACC	MCC
	RA_AC5_30	0.910	0.943	0.927	0.854
	RA_AC5_40	0.907	0.947	0.927	0.855
	WR_AC5_26	0.910	0.949	0.929	0.859
	RA_ALL_10*	0.915	0.936	0.925	0.851
	RA_ALL_20	0.911	0.938	0.925	0.850
	RA_ALL_30	0.918	0.954	0.936	0.872
	RA_ALL_40	0.918	0.958	0.938	0.876
	WR_ALL_25	0.914	0.952	0.933	0.867
Antiparasitario	RA_NO_10	0.778	0.778	0.778	0.556
	RA_NO_20	0.747	0.828	0.788	0.578
	RA_NO_30*	0.808	0.859	0.833	0.668
	RA_NO_40	0.808	0.838	0.823	0.647
	WR_NO_17*	0.818	**0.879**	**0.848**	**0.698**
	RA_ES_10	0.727	0.737	0.732	0.465
	RA_ES_20	0.758	0.737	0.747	0.495
	RA_ES_30	0.768	0.808	0.788	0.576
	RA_ES_40	0.778	0.818	0.798	0.596
	WR_ES_13	**0.838**	0.778	0.808	0.617
	RA_KH_10	0.697	0.727	0.712	0.424
	RA_KH_20	0.788	0.818	0.803	0.606
	RA_KH_30	0.828	0.798	0.813	0.627
	RA_KH_40	0.789	0.818	0.803	0.606
	WR_KH_20	0.808	0.859	0.833	0.668
	RA_AC2_10	0.679	0.778	0.737	0.476
	RA_AC2_20	0.737	0.758	0.747	0.495
	RA_AC2_30	0.788	0.818	0.803	0.606
	RA_AC2_40	0.768	0.808	0.788	0.576
	WR_AC2_17	0.818	0.808	0.813	0.626
	RA_AC5_10	0.677	0.727	0.702	0.405
	RA_AC5_20	0.727	0.778	0.753	0.506
	RA_AC5_30	0.747	0.788	0.768	0.536
	RA_AC5_40	0.758	0.798	0.778	0.556
	WR_AC5_20*	0.798	0.838	0.818	0.637
	RA_ALL_10	0.707	0.778	0.742	0.486
	RA_ALL_20	0.747	0.788	0.768	0.536
	RA_ALL_30	0.758	0.808	0.783	0.566
	RA_ALL_40	0.758	0.818	0.788	0.577
	WR_ALL_27*	0.788	0.818	0.803	0.606
Antiviral	RA_NO_10	0.728	0.808	0.768	0.538
	RA_NO_20	0.762	0.849	0.805	0.613
	RA_NO_30	0.765	0.854	0.809	0.621
	RA_NO_40	0.773	**0.863**	0.818	0.638
	WR_NO_28*	**0.776**	**0.863**	**0.820**	**0.642**
	RA_ES_10	0.732	0.785	0.759	0.518
	RA_ES_20	0.757	0.816	0.787	0.574
	RA_ES_30	0.763	0.825	0.794	0.589
	RA_ES_40	0.775	0.838	0.806	0.615
	WR_ES_29*	0.775	0.851	0.813	0.627
	RA_KH_10	0.681	0.761	0.721	0.443
	RA_KH_20	0.713	0.800	0.756	0.515
	RA_KH_30	0.726	0.813	0.770	0.541
	RA_KH_40	0.735	0.822	0.779	0.560
	WR_KH_23*	0.742	0.805	0.774	0.594
	RA_AC2_10	0.703	0.796	0.749	0.501

Continúa en la siguiente página

Tabla 6 – Continuación de la página anterior

Conjunto de entrenamiento	Modelo[a]	SN	SP	ACC	MCC
	RA_AC2_20	0.718	0.811	0.764	0.531
	RA_AC2_30	0.738	0.828	0.783	0.568
	RA_AC2_40	0.748	0.837	0.793	0.587
	WR_AC2_29	0.749	0.829	0.789	0.580
	RA_AC5_10	0.676	0.727	0.701	0.403
	RA_AC5_20	0.689	0.784	0.737	0.475
	RA_AC5_30	0.715	0.823	0.769	0.541
	RA_AC5_40	0.738	0.838	0.788	0.578
	WR_AC5_30	0.737	0.838	0.788	0.578
	RA_ALL_10	0.732	0.785	0.759	0.518
	RA_ALL_20	0.760	0.827	0.793	0.588
	RA_ALL_30	0.765	0.830	0.798	0.596
	RA_ALL_40*	0.762	0.840	0.801	0.603
	WR_ALL_25	0.760	0.836	0.798	0.597

[a] Los modelos de clasificación llevan por nombre la representación usada para su entrenamiento.

* Modelos seleccionados para el ensamble.

Tabla 7. Resultados en las métricas de desempeño, en un proceso de validación cruzada de 10 pliegues, de los modelos basados en descriptores moleculares con los que se entrenaron los clasificadores seleccionados para el ensamble de cada actividad. Los modelos de clasificación llevan por nombre la representación usada para su entrenamiento.

Conjunto de entrenamiento	Modelo	SN	SP	ACC	MCC
Antibacteriano	WR_EM_61	0.894	0.958	0.926	0.854
Antifúngico	WR_EM_21	0.918	0.959	0.938	0.877
Antiparasitario	WR_EM_43	0.808	0.869	0.838	0.678
Antiviral	WR_EM_69	0.779	0.874	0.826	0.656

La nomenclatura EM (ensamble), hace referencia a la metodología usada para construir cada representación.

Los modelos WR_EM_61 y WR_EM_69 obtuvieron las mejores métricas de desempeño para las actividades antibacteriana y antiviral, con un MCC de 0.854 y 0.656, mejorando esta métrica en 3.8 % y 1.4 % respecto a los modelos mejor evaluados de los construidos previamente en este trabajo para estas clases. Por otro lado, Los modelos WR_EM_21 y WR_EM_43 mostraron métricas de desempeño comparables respecto a los mejores modelos para actividad antifúngica y antiparasitaria, con una diferencia de 1.6 % y 2 % en el valor de MCC, respectivamente. Por lo anterior, se seleccionaron los cuatro modelos para análisis posteriores.

Además de los clasificadores construidos basados en descriptores moleculares de ProtDCal, se entrenaron clasificadores con una representación basada en descriptores de PseAAC. La Tabla 8 muestra las métricas de desempeño obtenidas con estos modelos.

Tabla 8. Resultados en las métricas de desempeño, en un proceso de validación cruzada de 10 pliegues, de los modelos basados en descriptores moleculares de PseACC.

Conjunto de entrenamiento	Modelo	SN	SP	ACC	MCC
Antibacteriano	PseAAC	0.895	0.944	0.919	0.840
Antifúngico	PseAAC	0.910	0.963	0.936	0.874
Antiparasitario	PseAAC	0.838	0.889	0.864	0.728
Antiviral	PseAAC	0.789	0.877	0.833	0.669

Los clasificadores basados en PseAAC para las actividades antiparasitaria y antiviral obtuvieron mejores métricas de desempeño comparados con los modelos mejor evaluados construidos con descritores de ProtDCal, para estas actividades, con un MCC de 0.728 y 0.669, mejorando el rendimiento en esta métrica por 3 % y 1.3 %, respectivamente. Por otra parte, los modelos para las actividades antibacteriana y antifúngica, mostraron un rendimiento comparable a los entrenados con las representaciones de ProtDCal, con una diferencia de 1.4 % y 1.9 %, respectivamente, en el valor del MCC. Por ello, estos cuatro modelos también se seleccionaron para futuras comparaciones.

Para el caso de la actividad antimicrobiana, solo se construyeron dos modelos de clasificación bajo la metodología propuesta en la sección 3.3. La Tabla 9 contiene los resultados observados en las métricas de desempeño medidas bajo una validación cruzada de 10 pliegues. La evaluación muestra que ambas representaciones utilizadas son comparables para discriminar entre casos positivos y negativos, con una diferencia en el valor de MCC de 0.007.

Tabla 9. Resultados en las métricas de desempeño, en un proceso de validación cruzada de 10 pliegues, de los modelos para clasificar actividad antimicrobiana. Los clasificadores llevan por nombre la respresentación usada para su entrenamiento.

Conjunto de entrenamiento	Modelo	SN	SP	ACC	MCC
Antimicrobiano	PseAAC	0.897	0.894	0.896	0.785
	WR_AMP_131	0.903	0.894	0.899	0.792

4.3. Desempeño de los modelos profundos de clasificación

Para construir los clasificadores basados en algoritmos de aprendizaje profundo, se llevó a cabo una optimización de hiperparámetros de la arquitectura descrita en la sección 3.4, la Tabla 10 exhibe los hiperparámetros encontrados para cada actividad.

Una vez optimizados los hiperparámetros de la red propuesta, se llevó a cabo un proceso de validación cruzada de 10 pliegues sobre los conjuntos de entrenamiento

Tabla 10. Hiperparámetros encontrados bajo un proceso de búsqueda en rejilla.

Actividad	Longitud del incrustamiento	Tipo de Unidad	No. Unidades Recurrentes	No. Unidades Densas	Tasa Dropout	Optimizador
Antimicrobiana	64	GRU	256	32	0.3	adam
Antibacteriana	256	GRU	256	32	0.3	adam
Antifúngica	256	LSTM	32	256	0.3	adam
Antiparasitaria	256	LSTM	64	256	0.5	adam
Antiviral	256	GRU	256	128	0.3	adam

de cada actividad, los modelos fueron compilados ajustando el parámetro *epochs*=10 y *loss*='binary_crossentropy'. La Tabla 11 muestra los resultados obtenidos.

Tabla 11. Resultados en las métricas de desempeño, en un proceso de validación cruzada de 10 pliegues, de los modelos de clasificación basados en DNNs.

Conjunto de entrenamiento	Nombre dado al modelo	SN	SP	ACC	MCC
Antimicrobiano	DeepRGM	0.856	0.888	0.869	0.734
Antibacteriano	DeepRGB	0.883	0.943	0.913	0.829
Antifúngico	DeepRLF	0.886	0.959	0.922	0.848
Antiparasitario	DeepRLP	0.829	0.840	0.834	0.682
Antiviral	DeepRGV	0.727	0.848	0.788	0.583

4.4. Comparación de los modelos clásicos y profundos de clasificación

Para efectuar una comparación justa entre los clasificadores clásicos y profundos, primero, los modelos se evaluaron bajo un proceso de validación por reserva repetido como se menciona en la sección 3.5. Para el caso de los modelos clásicos, se consideraron dos clasificadores por actividad, con el objetivo de evaluar las representaciones basadas en descriptores moleculares de PseAAC y ProtDCal. La Tabla 12 resume los resultados obtenidos en este proceso de validación. Los experimentos se llevaron a cabo en un equipo de cómputo con sistema operativo Windows 10, 16 GB de memoria RAM y procesador Intel Core i7.

Los resultados muestran que los clasificadores entrenados con descriptores moleculares de ProtDCal mejoran, en promedio, en un 1.4 % los resultados en el MCC a los modelos entrenados con descriptores de PseAAC para todos los conjuntos, excepto para el conjunto antiviral. Los modelos clásicos mejor evaluados superan en un 1.1 %, 3.4 % y 11.9 % en el valor de MCC, a los modelos profundos, en las actividades antimicrobiana, antifúngica y antiviral, respectivamente. Por el contrario, los modelos profundos mejoran el valor del MCC en un 1.5 % y 1.4 %, en las clases antibacteriana y antiparasitaria a los modelos clásicos.

Tabla 12. Resultados en las métricas de desempeño bajo un proceso de validación por reserva, repetido 30 veces, las métricas representan el promedio obtenido por cada modelo en este proceso. Los valores entre paréntesis representan la desviación estándar observada.

Conjunto	Modelo	SN	SP	ACC	MCC
Antimicrobiano	PseAAC	0.895(±0.006)	0.889(±0.006)	0.893(±0.005)	0.779(±0.01)
	WR_AMP_131	**0.904**(±0.005)	0.890(±0.006)	**0.898**(±0.004)	**0.789**(±0.008)
	DeepRGM	0.885(±0.019)	**0.901**(±0.027)	0.891(±0.006)	0.778(±0.012)
Antibacteriano	PseAAC	0.893(±0.007)	0.943(±0.005)	0.918(±0.004)	0.836(±0.008)
	WR_EM_61	0.892(±0.009)	**0.955**(±0.008)	0.924(±0.005)	0.850(±0.009)
	DeepRGB	**0.913**(±0.014)	0.951(±0.019)	**0.932**(±0.007)	**0.865**(±0.014)
Antifúngico	PseAAC	0.917(±0.023)	0.958(±0.016)	0.937(±0.016)	0.876(±0.031)
	WR_EM_21	**0.928**(±0.021)	**0.962**(±0.019)	**0.945**(±0.014)	**0.890**(±0.029)
	DeepRLF	0.908(±0.027)	0.946(±0.025)	0.927(±0.015)	0.856(±0.029)
Antiparasitario	PseAAC	0.835(±0.092)	0.848(±0.076)	0.842(±0.06)	0.688(±0.12)
	WR_EM_43	0.825(±0.125)	**0.872**(±0.063)	0.848(±0.062)	0.705(±0.012)
	DeepRLP	**0.845**(±0.092)	0.867(±0.075)	**0.856**(±0.045)	**0.719**(±0.089)
Antiviral	PseAAC	**0.788**(±0.02)	**0.873**(±0.014)	**0.830**(±0.013)	**0.663**(±0.025)
	WR_EM_69	0.776(±0.017)	0.869(±0.017)	0.822(±0.011)	0.648(±0.022)
	DeepRGV	0.709(±0.053)	0.828(±0.064)	0.768(±0.019)	0.544(±0.04)

Siguiendo con las comparaciones, se evaluó a los clasificadores clásicos y profundos usando los conjuntos de prueba y externos descritos en las secciones 3.1.1 y 3.1.2, después de haber sido entrenados con sus respectivos conjuntos de entrenamiento; las tablas 13 y 14 resumen los resultados obtenidos. Adicionalmente, los modelos para actividad antimicrobiana se evaluaron usando el conjunto externo extraído de los datos propuestos por Gabere y Noble (2017), la Tabla 15 muestra los resultados registrados por cada uno de estos.

Es importante mencionar que se incorporó a las comparaciones el modelo de ensamble construído para cada actividad (antibacteriana, antifúngica, antiparasitaria y antiviral), sin embargo, los resultados obtenidos no mostraron mejorar el desempeño de los modelos individuales, debido a esto, y a que la complejidad para evaluar dichos modelos (sobre todo en el proceso de validación por reserva), se decidió no continuar con su evaluación. Cabe notar que para obtener una mejor idea del comportamiento de estos modelos se deben realizar todas las comparaciones propuestas. Los resultados obtenidos al medir el rendimiento de estos clasificadores en los conjuntos de prueba y externos se pueden encontrar en el Anexo C, Tabla 35.

Al medir la capacidad de generalización de los modelos en los conjuntos de prueba, se encontró que los clasificadores basados en algoritmos clásicos mejoran el valor de MCC para las clases antifúngica y antiviral, en un 4 % y 19.4 %, respectivamente, comparados a los modelos profundos. No obstante, los modelos profundos mejoraron el

Tabla 13. Resultados en las métricas de desempeño al evaluar los modelos usando los conjuntos de prueba.

Conjunto de prueba	Modelo	SN	SP	ACC	MCC
Antimicrobiano	PseAAC	0.872	0.902	0.887	0.774
	WR_AMP_131	**0.883**	0.897	0.890	0.780
	DeepRGM	0.879	**0.924**	**0.901**	**0.803**
Antibacteriano	PseAAC	0.896	0.949	0.922	0.846
	WR_EM_61	0.888	**0.960**	0.924	0.851
	DeepRGB	**0.922**	0.958	**0.940**	**0.880**
Antifúngico	PseAAC	0.898	**0.977**	**0.937**	**0.877**
	WR_EM_21	0.893	0.930	0.912	0.824
	DeepRGB	**0.926**	0.912	0.919	0.837
Antiparasitario	PseAAC	0.774	0.839	0.806	0.614
	WR_EM_43	0.774	0.710	0.742	0.485
	DeepRGB	**0.839**	**0.839**	**0.839**	**0.677**
Antiviral	PseAAC	**0.777**	**0.900**	**0.839**	**0.683**
	WR_EM_69	0.751	0.876	0.814	0.633
	DeepRGB	0.612	0.862	0.737	0.489

Tabla 14. Resultados en las métricas de desempeño al evaluar los modelos usando los conjuntos externos.

Actividad	Modelo	Conjunto externo 1	SN	Conjunto externo 2	SN
Antimicrobiana	PseAAC	amp_external	0.929	biopep_amp	**0.759**
	WR_AMP_131		**0.953**		0.739
	DeepRGM		0.948		0.732
Antibacteriana	PseAAC	bacterial_fungal	0.866	biopep_bacterial	0.756
	WR_AMP_131		0.882		0.756
	DeepRGB		**0.907**		**0.796**
Antifúngica	PseAAC	bacterial_fungal	0.648	biopep_fungal	**0.787**
	WR_AMP_131		0.673		0.723
	DeepRGB		**0.761**		0.766
Antiparasitaria	PseAAC	parasitic_bacterial	0.752	–	–
	WR_AMP_131		**0.837**		–
	DeepRGB		0.766		–
Antiviral	PseAAC	fungal_viral	0.643	biopep_viral	0.200
	WR_AMP_131		0.689		0.300
	DeepRGB		**0.733**		**0.400**

MCC para las clases antimicrobiana, antibacteriana y antiparasitaria en un 2.3 %, 2.9 % y 6.3 %, a correspondencia. Por otra parte, los modelos profundos tuvieron un desempeño superior o comparable en los conjuntos externos para las actividades antibacteriana, antifúngica y antiviral, por el contrario, los clasificadores clásicos mostraron un mejor rendimiento en los conjuntos externos para las actividades antimicrobiana y antiparasitaria.

4.4.1. Comparación con modelos del estado del arte

Además de las comparaciones realizadas entre los modelos propuestos, estos se compararon con métodos reportados en la literatura (ver sección 3.6). Para llevar a

Tabla 15. Resultados en las métricas de desempeño al evaluar los modelos para actividad antimicrobiana usando el conjunto externo extraído de los datos propuestos por Gabere y Noble (2017).

Conjunto	Modelo	SN	SP	ACC	MCC
Externo Gabere y Noble	PseAAC	0.924	**0.866**	0.895	0.792
	WR_AMP_131	**0.966**	0.845	**0.905**	**0.816**
	DeepRGM	0.926	0.829	0.877	0.758

cabo una comparación justa, se eliminaron de los conjuntos de prueba y externos las secuencias contenidas en los conjuntos de entrenamiento de los modelos del estado del arte.

Las tablas 16 y 17 muestran la cantidad de secuencias en los conjuntos antes y después de filtrarlos. Los conjuntos externos extraídos de BIOPEP-UWM (Minkiewicz *et al.*, 2019) y de Gabere y Noble (2017), no contenían intersección con los conjuntos de entrenamiento de la literatura por ello mantuvieron su misma cardinalidad.

Tabla 16. Número de secuencias en los conjuntos de prueba antes y después de eliminar intersección con los conjuntos de entrenamiento de la literatura.

Conjunto	Prueba antes		Prueba después	
	Casos Positivos	Casos Negativos	Casos Positivos	Casos Negativos
Antimicrobiano	2564	2564	2323	2564
Antibacteriano	1695	1695	1541	1695
Antifúngico	215	215	103	215
Antiparasitario	31	31	26	31
Antiviral	623	623	272	623

Tabla 17. Número de secuencias en los conjuntos externos antes y después de eliminar intersección con los conjuntos de entrenamiento de la literatura.

Conjunto	No. de secuencias antes	No. de secuencias después
amp_external	3347	2935
bacterial_fungal	4757	2735
bacterial_fungal+	4757	2912
parasitic_bacterial	411	309
fungal_viral	1230	689

\+ Se mantuvieron dos versiones del conjunto *bacterial_fungal*, una para evaluar los modelos para la clase antibacteriana y otro para evaluar la clase antifúngica.

Una vez reducidos los conjuntos de datos, se evaluaron con estos tanto los clasificadores propuestos en este trabajo como los reportados en la literatura; la Tabla 18 muestra las métricas de desempeño obtenidas sobre los conjuntos de prueba. De la misma manera, las tablas 19 y 20, presentan la sensibilidad alcanzada por cada modelo en los conjuntos externos. Y, por último, la Tabla 21, resume los resultados obtenidos por los modelos de clasificación de actividad antimicrobiana en el conjunto extraido de Gabere y Noble (2017).

Tabla 18. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades estudiadas usando los conjuntos de prueba respectivos.

Conjunto	Modelo	P/N	SN	SP	ACC	MCC
Antimicrobiano	PseAAC	14951/9767	0.882	0.902	0.893	0.785
	WR_AMP_131	14951/9767	**0.896**	0.897	0.896	0.792
	DeepRGM	14951/9767	0.890	0.924	**0.908**	**0.815**
	iAMP-2L+	878/2405	0.647	0.848	0.752	0.507
	MLAMP+	878/2405	0.540	0.779	0.665	0.329
	AMPfun	1686/16428	0.806	0.137	0.455	-0.076
	AMPScanner	2021/2021	0.738	**0.948**	0.848	0.707
	ADAM*	–	0.822	0.550	0.680	0.385
	CAMPR3-SVM*	3010/4011	0.631	0.832	0.736	0.474
	CAMPR3-RF*	3010/4011	0.607	0.878	0.749	0.507
	CAMPR3-ANN*	3010/4011	0.632	0.860	0.752	0.508
	CAMPR3-DA*	3010/4011	0.610	0.853	0.737	0.479
Antibacteriano	PseAAC	6583/6583	0.897	0.949	0.924	0.849
	WR_EM_61	6583/6583	0.893	**0.961**	0.929	0.858
	DeepRGB	6583/6583	**0.927**	0.958	**0.943**	**0.886**
	PEPred	800/800	0.362	0.062	0.205	-0.609
	AMPfun	1930/1624	0.668	0.762	0.717	0.432
	iAMPpred	984/984	0.756	0.858	0.810	0.619
	MLAMP+	878	0.576	0.791	0.689	0.377
	iAMP-2L+	878	0.692	0.845	0.772	0.545
	ABP-Finder*	–	0.799	0.615	0.703	0.420
	AMPScanner	2021/2021	0.846	0.945	0.898	0.798
	ClassAMP-SVM*	454/908	0.463	0.736	0.606	0.208
Antifúngico	PseAAC	778/778	0.913	0.977	**0.956**	**0.899**
	WR_EM_21	778/778	0.903	0.930	0.921	0.823
	DeepRLF	778/778	**0.932**	0.912	0.918	0.822
	AntiFP_m1	1168/1168	0.019	**1.0**	0.682	0.115
	AntiFP_m2	1168/1168	0.641	0.991	0.877	0.721
	AntiFP_m3	1168/1168	0.097	**1.0**	0.708	0.260
	AMPfun	1912/1261	0.282	0.912	0.708	0.253
	iAMPpred	1384/1384	0.825	0.916	0.887	0.742
	MLAMP+	878	0.126	0.967	0.695	0.181
	iAMP-2L+	878	0.097	0.995	0.704	0.238
	ClassAMP-SVM*	61/122	0.262	0.428	0.374	-0.291
Antiparasitario	PseAAC	99/99	0.769	**0.839**	0.807	0.610
	WR_EM_43	99/99	0.808	0.710	0.754	0.516
	DeepRLP	99/99	**0.846**	**0.839**	**0.842**	**0.683**
	AMPfun	140/700	0.538	0.710	0.632	0.252
Antiviral	PseAAC	2321/2321	0.809	0.900	**0.873**	**0.702**
	WR_EM_69	2321/2321	0.768	0.876	0.844	0.636
	DeepRGV	2321/2321	0.585	0.862	0.778	0.461
	PEPred	544/407	0.496	0.332	0.382	-0.162
	AMPfun	1400/2451	**0.860**	0.255	0.439	0.128
	iAMPpred	739/739	0.515	0.860	0.755	0.396
	MLAMP+	878	0.004	**1.0**	0.697	0.051
	iAMP-2L+	878	0.037	**1.0**	0.707	0.161
	ClassAMP-SVM*	54/108	0.173	0.811	0.617	-0.02

La columna **P/N** muestra el número de secuencias usadas en el conjunto de entrenamiento de cada modelo. Para los modelos marcados con un * no se obtuvo su conjunto de entrenamiento. Los modelos marcados con el signo + siguen un enfoque de clasificación multietiqueta.

Tabla 19. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades estudiadas usando los conjuntos externos.

Actividad	Conjunto	Modelo	P/N	SN
Antimicrobiano	amp_external	PseAAC	14951/9767	0.926
		WR_AMP_131	14951/9767	**0.950**
		DeepRGM	14951/9767	0.945
		iAMP-2L+	878/2405	0.777
		MLAMP+	878/2405	0.639
		AMPfun	1686/16428	0.816
		AMPScanner	2021/2021	0.905
		ADAM*	–	0.904
		CAMPR3-SVM	3010/4011	0.774
		CAMPR3-RF	3010/4011	0.769
		CAMPR3-ANN	3010/4011	0.761
		CAMPR3-DA	3010/4011	0.775
Antibacteriano	bacterial_fungal	PseAAC	6583/6583	0.872
		WR_EM_61	6583/6583	0.884
		DeepRGB	6583/6583	**0.911**
		PEPred	800/800	0.306
		AMPfun	1930/1624	0.745
		iAMPpred	984/984	0.781
		MLAMP+	878	0.604
		iAMP-2L+	878	0.728
		ABP-Finder*	–	0.828
		AMPScanner	2021/2021	0.900
		ClassAMP-SVM*	454/908	0.458
Antifúngico	bacterial_fungal+	PseAAC	778/778	0.662
		WR_EM_21	778/778	0.660
		DeepRLF	778/778	**0.756**
		AntiFP_m1	1168/1168	0.007
		AntiFP_m2	1168/1168	0.167
		AntiFP_m3	1168/1168	0.052
		AMPfun	1912/1261	0.525
		iAMPpred	1384/1384	0.649
		MLAMP+	878	0.187
		iAMP-2L+	878	0.276
		ClassAMP-SVM*	61/122	0.298
Antiparasitario	parasitic_bacterial	PseAAC	99/99	0.735
		WR_EM_43	99/99	**0.841**
		DeepRLP	99/99	0.735
		AMPfun	140/700	0.450
Antiviral	fungal_viral	PseAAC	2321/2321	0.637
		WR_EM_69	2321/2321	0.717
		DeepRGV	2321/2321	0.772
		PEPred	544/407	0.212
		AMPfun	1400/2451	0.260
		iAMPpred	739/739	**0.804**
		MLAMP+	878	0.090
		iAMP-2L+	878	0.151
		ClassAMP-SVM*	54/108	0.251

La columna **P/N** muestra el número de secuencias usadas en el conjunto de entrenamiento de cada modelo. Para los modelos marcados con un * no se obtuvo su conjunto de entrenamiento. Los modelos marcados con el signo + siguen un enfoque de clasificación multietiqueta.

Tabla 20. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades estudiadas usando los conjuntos externos extraídos de BIOPEP-UWM (Minkiewiczet al., 2019).

Actividad	Conjunto	Modelo	P/N	SN
Antimicrobiano	biopep_amp	PseAAC	14951/9767	0.759
		WR_AMP_131	14951/9767	0.739
		DeepRGM	14951/9767	0.732
		iAMP-2L+	878/2405	0.825
		MLAMP+	878/2405	0.642
		AMPfun	1686/16428	**0.996**
		AMPScanner	2021/2021	0.689
		ADAM*	–	0.895
		CAMPR3-SVM	3010/4011	0.700
		CAMPR3-RF	3010/4011	0.681
		CAMPR3-ANN	3010/4011	0.665
		CAMPR3-DA	3010/4011	0.669
Antibacteriano	biopep_bacterial	PseAAC	6583/6583	0.756
		WR_EM_61	6583/6583	0.756
		DeepRGB	6583/6583	0.796
		PEPred	800/800	0.239
		AMPfun	1930/1624	0.810
		iAMPpred	984/984	0.786
		MLAMP+	878	0.692
		iAMP-2L+	878	**0.816**
		ABP-Finder*	–	0.692
		AMPScanner	2021/2021	0.776
		ClassAMP-SVM*	454/908	0.303
Antifúngico	biopep_fungal	PseAAC	778/778	**0.787**
		WR_EM_21	778/778	0.723
		DeepRLF	778/778	0.766
		AntiFP_m1	1168/1168	0.0
		AntiFP_m2	1168/1168	0.0
		AntiFP_m3	1168/1168	0.0
		AMPfun	1912/1261	0.298
		iAMPpred	1384/1384	0.383
		MLAMP+	878	0.064
		iAMP-2L+	878	0.0
		ClassAMP-SVM*	61/122	0.383
Antiviral	biopep_viral	PseAAC	2321/2321	0.200
		WR_EM_69	2321/2321	0.300
		DeepRGV	2321/2321	0.400
		PEPred	544/407	**1.0**
		AMPfun	1400/2451	**1.0**
		iAMPpred	739/739	0.200
		MLAMP+	878	0.0
		iAMP-2L+	878	0.0
		ClassAMP-SVM*	54/108	0.300

La columna **P/N** muestra el número de secuencias usadas en el conjunto de entrenamiento de cada modelo. Para los modelos marcados con un * no se obtuvo su conjunto de entrenamiento. Los modelos marcados con el signo + siguen un enfoque de clasificación multietiqueta.

En las tablas 18 a 20, se puede apreciar el número de secuencias positivas y negativas con las que fueron entrenados los clasificadores evaluados, sin embargo, a

Tabla 21. Comparación en las métricas de desempeño al evaluar los modelos para actividad antimicrobiana usando el conjunto extraido de los datos propuestos por Gabere y Noble (2017).

Conjunto	Modelo	SN	SP	ACC	MCC
Externo Gabere y Noble	PseAAC	0.924	0.866	0.895	0.792
	WR_AMP_131	**0.966**	0.845	0.905	0.816
	DeepRGM	0.926	0.829	0.877	0.758
	iAMP-2L	0.803	0.863	0.833	0.667
	MLAMP	0.637	0.782	0.709	0.423
	AMPfun	–	–	–	–
	AMPScanner	0.958	**0.881**	**0.920**	**0.842**
	ADAM	–	–	–	–
	CAMPR3-SVM	0.921	0.780	0.850	0.707
	CAMPR3-RF	0.946	0.787	0.866	0.742
	CAMPR3-ANN	0.816	0.841	0.829	0.657
	CAMPR3-DA	0.890	0.780	0.835	0.674

Para los modelos marcados con un * no se obtuvo su conjunto de entrenamiento. Algunos modelos no generaron reultados, por lo tanto, las métricas de estos están marcadas con guiones.

diferencia de los otros métodos, para MLAMP e iAMP-2L (marcados con el signo +) solo se presenta una cantidad (878), esto se debe a que estos clasificadores siguen un enfoque multietiqueta, es decir, las secuencias utilizadas pueden tener asociadas más de una función biológica, por lo tanto es difícil establecer la proporción de positivos/negativos.

Las evaluaciones muestran que los modelos propuestos en esta investigación tienen un mejor desempeño sobre el conjunto de prueba para todas las actividades estudiadas (ver Tabla 18), mientras que presentan un rendimiento de comparable a superior en los conjuntos externos, comparado con el rendimiento de los modelos reportados en la literatura (ver Tabla 19).

En la Tabla 20, se puede apreciar que los modelos propuestos en esta investigación tienen un rendimiento menor en las actividades antimicrobiana, antibacteriana y antiviral comparado con los modelos AMPfun, iAMP-2L y PEPred, respectivamente, pero muestran un mejor desempeño en la clase antifúngica comparado con los otros métodos del estado del arte.

Las comparaciones anteriores podrían no ser del todo justas, esto debido al problema de falsos negativos que exhiben los clasificadores jerárquicos Para considerar dicho inconveniente se calcularon las métricas de desempeño utilizando solo las secuencias predichas como AMP, como se menciona en la sección 3.6.1. La Tabla 22 muestra el número de secuencias predichas como AMP por los clasificadores jerárquicos de la

Tabla 22. Número de secuencias en los conjuntos positivos de prueba y en conjuntos externos originales y predichos como AMP por el primer filtro de servidores jerárquicos. El valor entre paréntesis representa el porcentaje de secuencias predichas como AMP.

Conjunto	Modelo	Prueba		Externos	
		Positivos Original	Positivos Predichos	Positivos Original	Positivos Predichos
Antibacteriano	AMPfun	1541	1170(76%)	2735	2197(80%)
	MLAMP	1541	889(58%)	2735	1712(63%)
	iAMP-2L	1541	1072(70%)	2735	2084(76%)
	PseAAC	1541	1414(92%)	2735	2524(92%)
	WR_AMP_131	1541	1433(93%)	2735	2592(95%)
Antifúngico	AMPfun	103	42(41%)	2912	2374(82%)
	MLAMP	103	29(28%)	2912	1859(64%)
	iAMP-2L	103	33(32%)	2912	2247(77%)
	PseAAC	103	101(98%)	2912	2691(92%)
	WR_AMP_131	103	99(96%)	2912	2763(95%)
Antiparasitario	AMPfun	26	24(92%)	309	286(93%)
	PseAAC	26	22(85%)	309	301(97%)
	WR_AMP_131	26	22(85%)	309	304(98%)
Antiviral	AMPfun	272	259(95%)	689	667(97%)
	MLAMP	272	134(49%)	689	534(78%)
	iAMP-2L	272	139(51%)	689	651(94%)
	PseAAC	272	205(75%)	689	662(96%)
	WR_AMP_131	272	211(78%)	689	676(98%)

literatura y por los modelos clásicos propuestos en este trabajo.

De la Tabla 22 se puede destacar que los modelos de clasificación de AMPs propuestos en esta investigación muestran una mayor sensibilidad (porcentaje de secuencias clasificadas correctamente) en la mayoría de los conjuntos.

A continuación, las tablas 23 a la 27, resumen los resultados obtenidos al calcular las métricas de desempeño bajo el enfoque jerárquico utilizado por métodos reportados en la literatura, así mismo, incluyendo los modelos propuestos en esta investigación sobre los conjuntos de prueba y externos.

Tabla 23. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral usando los conjuntos de prueba y los conjuntos externos (bacterial_fungal, funga_viral y parasitic_bacterial) en los que solo se contemplan las secuencias que fueron predichas como AMP por el servidor AMPfun.

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriano	PseAAC	0.890	0.949	0.925	0.844	0.876
	WR_EM_61	0.882	**0.961**	0.929	0.852	0.885
	DeepRGB	**0.910**	0.958	**0.938**	**0.872**	0.901
	PEPred	0.227	0.062	0.129	-0.731	0.187
	AMPfun	0.879	0.762	0.810	0.630	**0.927**
	iAMPpred	0.826	0.858	0.845	0.682	0.874
	MLAMP	0.745	0.791	0.772	0.533	0.748
	iAMP-2L	0.873	0.845	0.856	0.709	0.879
	ABP-Finder*	0.770	0.615	0.678	0.380	0.802
	AMPScanner	0.854	0.945	0.908	0.809	0.899
	ClassAMP-SVM*	0.384	0.736	0.592	0.127	0.374

Continúa en la siguiente página

Tabla 23 – Continuación de la página anterior

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antifúngico	PseAAC	0.929	0.977	**0.969**	**0.889**	0.764
	WR_EM_21	**0.952**	0.930	0.934	0.796	0.746
	DeepRLF	0.905	0.912	0.911	0.727	**0.806**
	AntiFP_m1	0.048	**1.00**	0.844	0.200	0.008
	AntiFP_m2	0.262	0.991	0.872	0.426	0.101
	AntiFP_m3	0.167	**1.00**	0.864	0.379	0.061
	AMPfun	0.690	0.912	0.875	0.571	0.644
	iAMPpred	0.714	0.916	0.883	0.598	0.714
	MLAMP	0.310	0.967	0.860	0.382	0.228
	iAMP-2L	0.238	0.995	0.872	0.426	0.332
	ClassAMP-SVM*	0.333	0.428	0.412	-0.177	0.336
Antiparasitario	PseAAC	0.750	**0.839**	0.800	0.592	0.752
	WR_EM_43	0.792	0.710	0.745	0.497	**0.836**
	DeepRLP	**0.833**	**0.839**	**0.836**	**0.669**	0.752
	AMPfun	0.583	0.710	0.655	0.295	0.486
Antiviral	PseAAC	0.807	0.900	**0.873**	**0.698**	0.648
	WR_EM_69	0.761	0.876	0.842	0.627	0.708
	DeepRGV	0.595	0.862	0.783	0.467	0.769
	PEPred	0.510	0.332	0.384	-0.148	0.219
	AMPfun	**0.903**	0.255	0.446	0.178	0.268
	iAMPpred	0.498	0.860	0.754	0.380	**0.805**
	MLAMP	0.004	**1.00**	0.707	0.052	0.093
	iAMP-2L	0.039	**1.00**	0.718	0.166	0.156
	ClassAMP-SVM*	0.178	0.811	0.625	-0.014	0.247

Para los modelos marcados con un * no se eliminó intersección con su conjunto de entrenamiento.

Tabla 24. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral usando los conjuntos de prueba y los conjuntos externos (bacterial_fungal, funga_viral y parasitic_bacterial) en los que solo se contemplan las secuencias que fueron predichas como AMP por el servidor MLAMP.

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriano	PseAAC	0.948	0.949	0.949	0.889	0.928
	WR_EM_61	0.945	**0.961**	0.955	0.902	0.933
	DeepRGB	0.962	0.958	**0.959**	**0.911**	0.949
	PEPred	0.160	0.062	0.096	-0.787	0.120
	AMPfun	0.937	0.774	0.830	0.677	0.957
	iAMPpred	0.875	0.858	0.864	0.713	0.925
	MLAMP	**0.998**	0.791	0.862	0.750	**0.965**
	iAMP-2L	0.931	0.845	0.875	0.747	0.910
	ABP-Finder*	0.805	0.615	0.680	0.400	0.831
	AMPScanner	0.913	0.945	0.934	0.855	0.942
	ClassAMP-SVM*	0.416	0.736	0.626	0.156	0.395
Antifúngico	PseAAC	**0.931**	0.977	**0.971**	**0.870**	0.800
	WR_EM_21	**0.931**	0.930	0.930	0.738	0.762
	DeepRLF	0.897	0.912	0.910	0.674	**0.821**
	AntiFP_m1	0.034	**1.00**	0.885	0.175	0.008
	AntiFP_m2	0.207	0.991	0.898	0.359	0.107
	AntiFP_m3	0.172	**1.00**	0.902	0.394	0.066
	AMPfun	0.724	0.912	0.889	0.556	0.684

Continúa en la siguiente página

Tabla 24 – Continuación de la página anterior

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
	iAMPpred	0.793	0.916	0.902	0.614	0.749
	MLAMP	0.448	0.967	0.906	0.490	0.293
	iAMP-2L	0.276	0.995	0.910	0.466	0.352
	ClassAMP-SVM*	0.310	0.428	0.414	-0.170	0.328
Antiviral	PseAAC	0.866	0.900	**0.894**	**0.690**	0.674
	WR_EM_69	0.813	0.876	0.865	0.612	0.708
	DeepRGV	0.731	0.862	0.839	0.528	0.800
	PEPred	0.493	0.332	0.361	-0.139	0.185
	AMPfun	**0.910**	0.255	0.371	0.151	0.273
	iAMPpred	0.537	0.860	0.803	0.373	**0.828**
	MLAMP	0.007	**1.00**	0.824	0.078	0.116
	iAMP-2L	0.067	**1.00**	0.835	0.237	0.174
	ClassAMP-SVM*	0.216	0.811	0.705	0.026	0.238

Para los modelos marcados con un * no se eliminó intersección con su conjunto de entrenamiento.

Tabla 25. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral usando los conjuntos de prueba y los conjuntos externos (bacterial_fungal, funga_viral y parasitic_bacterial) en los que solo se contemplan las secuencias que fueron predichas como AMP por el servidor iAMP-2L.

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriano	PseAAC	0.945	0.949	0.948	0.890	0.922
	WR_EM_61	0.943	**0.961**	0.954	0.903	0.930
	DeepRGB	0.965	0.958	**0.960**	**0.917**	0.947
	PEPred	0.162	0.062	0.101	-0.786	0.145
	AMPfun	0.898	0.774	0.822	0.655	0.929
	iAMPpred	0.900	0.858	0.875	0.745	0.924
	MLAMP	0.774	0.791	0.785	0.557	0.758
	iAMP-2L	**0.994**	0.845	0.903	0.818	**0.955**
	ABP-Finder*	0.800	0.615	0.687	0.406	0.815
	AMPScanner	0.923	0.945	0.937	0.867	0.943
	ClassAMP-SVM*	0.412	0.736	0.611	0.155	0.392
Antifúngico	PseAAC	**0.939**	0.977	**0.972**	**0.883**	0.798
	WR_EM_21	0.909	0.930	0.927	0.740	0.774
	DeepRLF	**0.939**	0.912	0.915	0.720	**0.832**
	AntiFP_m1	0.061	**1.00**	0.875	0.230	0.008
	AntiFP_m2	0.273	0.991	0.895	0.435	0.118
	AntiFP_m3	0.212	**1.00**	0.895	0.435	0.065
	AMPfun	0.727	0.912	0.887	0.573	0.656
	iAMPpred	0.818	0.916	0.903	0.647	0.757
	MLAMP	0.364	0.967	0.887	0.423	0.230
	iAMP-2L	0.303	0.995	0.903	0.492	0.358
	ClassAMP-SVM*	0.303	0.428	0.411	-0.183	0.319
Antiviral	PseAAC	0.899	0.900	**0.900**	**0.718**	0.662
	WR_EM_69	0.856	0.876	0.873	0.647	0.719
	DeepRGV	0.712	0.862	0.835	0.517	0.780
	PEPred	0.597	0.332	0.381	-0.057	0.215
	AMPfun	**0.914**	0.255	0.375	0.156	0.273
	iAMPpred	0.583	0.860	0.810	0.413	**0.817**
	MLAMP	0.007	**1.00**	0.819	0.077	0.094

Continúa en la siguiente página

Tabla 25 – Continuación de la página anterior

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
	iAMP-2L	0.072	**1.00**	0.831	0.244	0.160
	ClassAMP-SVM*	0.201	0.811	0.699	0.012	0.250

Para los modelos marcados con un * no se eliminó intersección con su conjunto de entrenamiento.

Tabla 26. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral usando los conjuntos de prueba y los conjuntos externos (bacterial_fungal, funga_viral y parasitic_bacterial) en los que solo se contemplan las secuencias que fueron predichas como AMP por el modelo propuesto PseAAC.

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriano	PseAAC	0.971	0.949	0.959	0.918	0.943
	WR_EM_61	0.955	**0.961**	0.958	0.916	0.941
	DeepRGB	**0.973**	0.958	**0.965**	**0.929**	**0.956**
	PEPred	0.316	0.062	0.178	-0.652	0.261
	AMPfun	0.694	0.762	0.731	0.460	0.777
	iAMPpred	0.803	0.858	0.833	0.663	0.829
	MLAMP	0.606	0.791	0.707	0.406	0.629
	iAMP-2L	0.731	0.845	0.793	0.581	0.763
	ABP-Finder*	0.820	0.615	0.708	0.439	0.841
	AMPScanner	0.909	0.945	0.929	0.856	0.947
	ClassAMP-SVM*	0.479	0.736	0.619	0.223	0.464
Antifúngico	PseAAC	0.931	0.977	**0.962**	**0.912**	0.709
	WR_EM_21	0.911	0.930	0.924	0.829	0.696
	DeepRLF	**0.941**	0.912	0.921	0.828	**0.796**
	AntiFP_m1	0.020	**1.00**	0.687	0.116	0.007
	AntiFP_m2	0.653	0.991	0.883	0.731	0.175
	AntiFP_m3	0.099	**1.00**	0.712	0.264	0.054
	AMPfun	0.287	0.912	0.712	0.258	0.557
	iAMPpred	0.822	0.916	0.886	0.738	0.689
	MLAMP	0.129	0.967	0.699	0.184	0.196
	iAMP-2L	0.099	0.995	0.709	0.240	0.293
	ClassAMP-SVM*	0.257	0.428	0.373	-0.294	0.288
Antiparasitario	PseAAC	0.864	**0.839**	0.849	0.695	0.754
	WR_EM_43	0.864	0.710	0.774	0.566	**0.854**
	DeepRLP	**0.909**	**0.839**	**0.868**	**0.738**	0.751
	AMPfun	0.636	0.710	0.679	0.344	0.458
Antiviral	PseAAC	**0.985**	0.900	**0.921**	**0.820**	0.660
	WR_EM_69	0.927	0.876	0.889	0.742	0.736
	DeepRGV	0.693	0.862	0.820	0.536	0.799
	PEPred	0.463	0.332	0.365	-0.181	0.199
	AMPfun	0.888	0.255	0.412	0.149	0.261
	iAMPpred	0.600	0.860	0.796	0.457	**0.828**
	MLAMP	0.005	**1.00**	0.754	0.061	0.094
	iAMP-2L	0.049	**1.00**	0.764	0.193	0.157
	ClassAMP-SVM*	0.185	0.811	0.656	-0.004	0.249

Para los modelos marcados con un * no se eliminó intersección con su conjunto de entrenamiento.

Tabla 27. Comparación en las métricas de desempeño al evaluar los modelos de clasificación para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral usando los conjuntos de prueba y los conjuntos externos (bacterial_fungal, funga_viral y parasitic_bacterial) en los que solo se contemplan las secuencias que fueron predichas como AMP por el modelo propuesto WR_AMP_131.

Conjunto	Modelo	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriano	PseAAC	0.948	0.949	0.949	0.896	0.911
	WR_EM_61	0.952	**0.961**	0.957	0.913	0.931
	DeepRGB	**0.972**	0.958	**0.964**	**0.928**	**0.948**
	PEPred	0.321	0.062	0.181	-0.647	0.275
	AMPfun	0.687	0.762	0.727	0.450	0.760
	iAMPpred	0.796	0.858	0.830	0.657	0.814
	MLAMP	0.599	0.791	0.703	0.399	0.615
	iAMP-2L	0.725	0.845	0.790	0.576	0.753
	ABP-Finder*	0.826	0.615	0.712	0.446	0.840
	AMPScanner	0.895	0.945	0.922	0.844	0.938
	ClassAMP-SVM*	0.480	0.736	0.619	0.224	0.470
Antifúngico	PseAAC	0.939	0.977	**0.965**	**0.919**	0.689
	WR_EM_21	0.909	0.930	0.924	0.827	0.685
	DeepRLF	**0.960**	0.912	0.927	0.842	**0.779**
	AntiFP_m1	0.020	**1.00**	0.691	0.118	0.007
	AntiFP_m2	0.667	0.991	0.889	0.742	0.174
	AntiFP_m3	0.101	**1.00**	0.717	0.267	0.052
	AMPfun	0.283	0.912	0.713	0.253	0.545
	iAMPpred	0.848	0.916	0.895	0.759	0.677
	MLAMP	0.121	0.967	0.701	0.173	0.191
	iAMP-2L	0.091	0.995	0.710	0.228	0.288
	ClassAMP-SVM*	0.253	0.428	0.373	-0.297	0.288
Antiparasitario	PseAAC	0.864	**0.839**	0.849	0.695	0.747
	WR_EM_43	**0.909**	0.710	0.792	0.613	**0.852**
	DeepRLP	**0.909**	**0.839**	**0.869**	**0.738**	0.747
	AMPfun	0.591	0.710	0.660	0.301	0.454
Antiviral	PseAAC	**0.957**	0.900	**0.915**	**0.802**	0.646
	WR_EM_69	0.929	0.876	0.890	0.746	0.726
	DeepRGV	0.673	0.862	0.814	0.522	0.786
	PEPred	0.445	0.332	0.361	-0.198	0.209
	AMPfun	0.858	0.255	0.408	0.117	0.259
	iAMPpred	0.611	0.860	0.797	0.468	**0.818**
	MLAMP	0.005	**1.00**	0.748	0.060	0.092
	iAMP-2L	0.047	**1.00**	0.759	0.189	0.154
	ClassAMP-SVM*	0.171	0.811	0.649	-0.021	0.249

Para los modelos marcados con un * no se eliminó intersección con su conjunto de entrenamiento.

En los resultados se puede observar que se mantiene un comportamiento similar en el desempeño de los modelos de clasificación para cada unas de las clases, siendo los modelos propuestos en este trabajo los que presentan mejor rendimiento en los conjuntos de prueba y un rendimiento de comparable a superior en los conjuntos externos respectivos.

4.5. Ejemplo de aplicación. Predicción de actividades antimicrobianas en un metagenoma de esponjas marinas del Parque Nacional Cabo Pulmo

Las secuencias del metagenoma de esponjas marinas se codificaron con las representaciones de descriptores moleculares de PseAAC (usada para todas las clases), WR_AMP_131 (usada para predecir actividad antimicrobiana), WR_EM_61 (actividad antibacteriana), WR_EM_21 (actividad antifúngica), WR_EM_43 (actividad antiprasitaria) y WR_EM_69 (actividad antiviral).

Posteriormente, se emplearon los clasificadores clásicos entrenados con estas representaciones para predecir su actividad. En la Tabla 28 se pueden apreciar las predicciones de cada modelo sobre cada una de las secuencias, todas aquellas que se predijeron con alguna actividad se marcan con un 1 y las que no presentaron la actividad según cada modelo se marcan con un 0.

Las secuencias 30, 32 y 52 fueron las que se evaluaron de forma positiva por la mayor cantidad de clasificadores, mientras que las secuencias 27, 28, 38, 42, 51, 56 y 63 fueron las que dieron negativas al estimar su potencial actividad en la mayoría de los modelos. Sería interesante estudiar las secuencias mencionadas anteriormente para validar la capacidad predictiva del modelo de manera experimental.

Tabla 28. Predicciones obtenidas al evaluar las secuencias extraídas de un metagenoma de esponjas marinas del Parque Nacional Cabo Pulmo, utilizando los clasificadores clásicos para cada una de las actividades estudiadas.

ID Secuencia	PseAAC AMP	WR_AMP_131	PseAAC ABP	WR_EM_61	PseAAC AFP	WR_EM_21	PseAAC APP	WR_EM_43	PseAAC AVP	WR_EM_69
Sequence_1	0	1	0	0	0	0	0	1	0	1
Sequence_2	1	1	0	0	0	0	1	1	1	0
Sequence_3	0	1	0	0	0	0	1	0	0	1
Sequence_4	1	1	0	0	0	0	1	1	1	0
Sequence_5	0	1	0	0	0	0	1	0	1	1
Sequence_6	0	1	0	0	0	0	1	1	1	1
Sequence_7	0	1	0	1	0	0	1	1	0	1
Sequence_8	0	1	0	0	0	0	1	1	0	1
Sequence_9	0	1	0	0	0	0	1	1	0	0
Sequence_10	0	1	0	0	0	0	0	1	1	0
Sequence_11	0	1	0	0	0	0	1	1	1	1
Sequence_12	1	1	0	0	0	0	0	1	1	1
Sequence_13	0	1	0	0	0	0	1	1	0	0
Sequence_14	0	1	0	0	0	0	0	1	1	1
Sequence_15	1	1	0	0	0	0	0	0	1	1
Sequence_16	0	1	0	0	0	0	1	1	1	0
Sequence_17	0	1	0	0	0	0	0	1	1	0
Sequence_18	0	0	0	0	0	0	1	1	1	0
Sequence_19	0	1	0	1	0	0	1	1	1	1
Sequence_20	0	1	0	0	0	0	0	1	1	1
Sequence_21	0	1	0	0	0	0	0	1	1	1
Sequence_22	0	1	0	0	0	0	0	1	1	1
Sequence_23	0	1	0	0	0	0	0	1	0	1
Sequence_24	1	1	0	0	0	0	0	1	1	1
Sequence_25	1	1	0	0	0	0	0	1	1	1
Sequence_26	0	1	0	1	0	0	0	1	0	1
Sequence_27	0	0	0	0	0	0	0	1	0	1
Sequence_28	0	0	0	0	0	0	0	0	0	0
Sequence_29	0	1	0	0	0	0	1	1	0	0
Sequence_30	1	1	1	1	1	0	1	1	1	1
Sequence_31	1	1	0	0	0	0	0	0	1	1
Sequence_32	1	1	1	1	0	0	1	1	0	1
Sequence_33	1	1	0	0	0	0	1	1	1	1
Sequence_34	1	1	0	0	0	0	0	1	1	0
Sequence_35	0	1	0	0	0	0	1	0	1	0
Sequence_36	0	1	0	0	0	0	1	1	1	1
Sequence_37	0	1	0	0	0	0	0	1	1	1
Sequence_38	0	1	0	0	0	0	0	0	0	1
Sequence_39	0	1	0	0	0	0	0	0	1	1
Sequence_40	0	1	1	0	0	0	1	1	1	1
Sequence_41	0	1	0	1	0	0	1	1	1	0
Sequence_42	0	0	0	0	0	0	0	1	1	0
Sequence_43	0	1	0	0	0	0	0	1	0	1
Sequence_44	1	1	1	0	1	0	1	0	1	1
Sequence_45	0	1	0	0	0	0	1	1	0	1
Sequence_46	0	1	0	0	0	0	0	0	1	1
Sequence_47	0	1	0	0	0	0	1	1	1	1
Sequence_48	1	1	0	0	0	0	1	1	1	1
Sequence_49	0	1	0	0	0	0	1	1	1	0
Sequence_50	1	1	0	0	0	0	1	1	1	1
Sequence_51	0	0	0	0	0	0	1	0	1	0
Sequence_52	1	1	1	1	0	0	1	1	1	1
Sequence_53	0	1	0	1	0	0	0	1	0	1
Sequence_54	0	1	0	0	0	0	1	1	0	1
Sequence_55	0	1	0	0	0	0	0	1	1	0
Sequence_56	0	0	0	0	0	0	0	1	0	0
Sequence_57	0	1	0	0	0	0	1	1	1	1
Sequence_58	0	1	0	0	0	0	0	1	1	0
Sequence_59	1	1	0	0	0	0	1	1	1	1
Sequence_60	0	0	0	0	0	0	1	1	0	1
Sequence_61	0	1	0	1	0	0	1	1	1	1
Sequence_62	1	1	0	0	0	0	1	1	1	1
Sequence_63	0	1	0	0	0	0	0	1	0	0
Sequence_64	0	1	0	0	0	0	1	1	1	1
Sequence_65	0	1	0	0	0	0	0	1	1	1
Sequence_66	0	0	0	0	0	0	0	1	1	1

Un 1 representa una predicción positiva y un 0 una predicción negativa. Los acrónimos ABP, AFP, APP y AVP hacen referencia a las actividades antibacteriana, antifúngica, antiparasitaria y antiviral, respectivamente.

Capítulo 5. Discusiones y conclusión

El presente capítulo discute los resultados de aplicar la metodología propuesta para lograr los objetivos de investigación. De la misma manera, expone las conclusiones a las que este trabajo condujo, así también, propone algunas ideas de investigación que se podrían explorar a futuro.

5.1. Discusiones

5.1.1. Conjuntos de datos

Para construir los clasificadores se propusieron cinco conjuntos de datos, uno para cada actividad estudiada. Una limitante al construir los conjuntos de datos es la falta de péptidos validados experimentalmente que carezcan de las actividades deseadas, por ello, se construyeron conjuntos de datos negativos, sin embargo, no se tiene la certeza de que los conjuntos construídos representen verdaderamente a los casos que no presentan las actividades buscadas. Lo ideal sería contar con un conjunto validado experimentalmente de secuencias sin las actividades analizadas.

Respecto a la calidad de los conjuntos propuestos, se puede observar en la sección 4.1, figuras 14 a 23, que los conjuntos antimicrobiano, antibacteriano y antiviral se podrían considerar de mayor calidad que los conjuntos de datos propuestos en la literatura basándonos en las métricas de parentesco y diversidad obtenidas y, además, considerando que muestrean mayormente el espacio de péptidos conocidos y validados experimentalmente. En contraste, los conjuntos antifúngico y antiparasitario, son de menor calidad, esto debido al poco parentesco entre casos positivos y negativos, lo que supone que estos conjuntos son fácilmente separables a nivel de secuencia y no se requieren de la complejidad de un algoritmo de aprendizaje de máquina para separar con buena exactitud a estos conjuntos. Sería interesante aplicar un método de agrupamiento basado en secuencia para confirmar la separabilidad de estos conjuntos.

Se encontró una limitante en la definición de la medida de parentesco (ver sección 2.2.3). Para ejemplificar este problema, suponga dos conjuntos de datos A y B, ambos con la misma cantidad de secuencias (5 positivas y 5 negativas), ahora considere los siguientes escenarios, $\forall a \in A_{pos},\ \exists a' \in A_{neg}$, tal que, $similitud(a, a') \geq \alpha$; por otra parte,

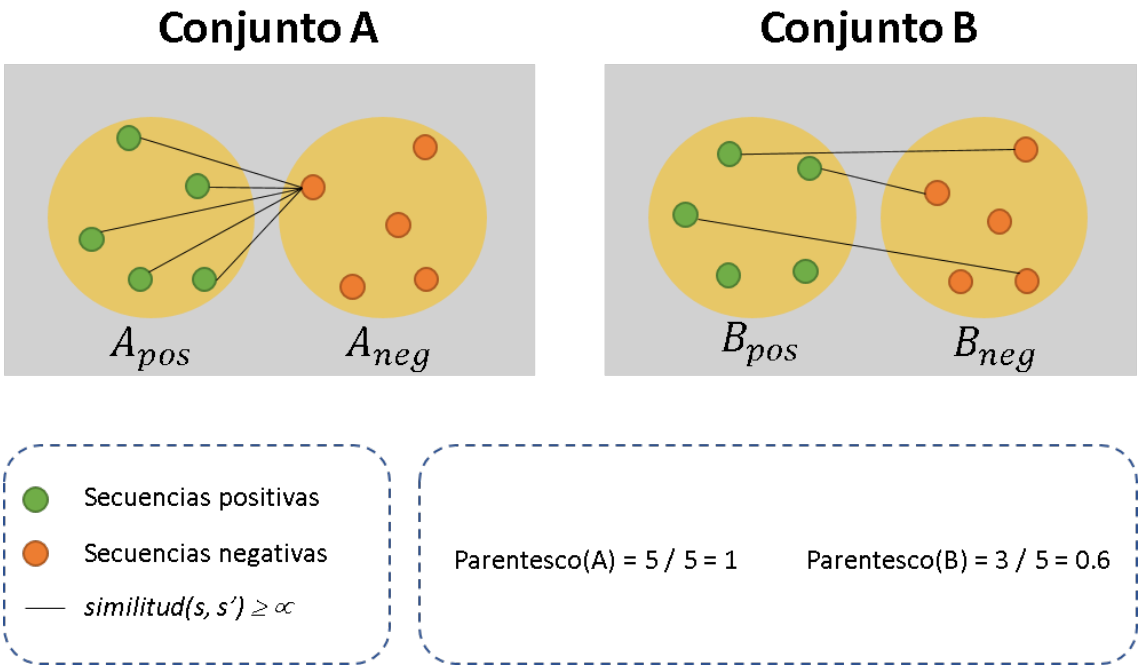

Figura 24. Ejemplo del problema encontrado en la definición de la medida de parentesco. Diagrama elaborado utilizando el software Microsoft PowerPoint versión 2016.

$\exists b_1, b_2, b_3 \in B_{pos}$ y $\exists b'_1, b'_2, b'_3 \in B_{neg}$, tal que, $similitud(b_i, b'_i) \geq \alpha,\ para\ i = 1, 2, 3$. Observe que **parentesco**$(A) = 1$ siguiendo la definición, mientras que **parentesco**$(B) = 0.6$; aunque la medida en el conjunto B es menor, tiene el comportamiento que deseamos, contrario a lo que sucede en el conjunto A. La Figura 24, muestra gráficamente el ejemplo abordado. De lo anterior, podemos decir que la medida de parentesco no es sensible al número de secuencias con las que se supera el valor α de similitud, pudiendo esto darnos una idea equivocada del nivel de separabilidad de los datos, por lo que se debe proponer una medida más robusta para medir dicha propiedad.

5.1.2. Comparación del desempeño de los modelos clásicos y profundos de clasificación

Los resultados contenidos en las tablas 12, 13 y 14 muestran que seleccionar descriptores moleculares independientes para cada clase no genera una diferencia considerable en el rendimiento de los modelos de clasificación clásicos, con descriptores tan generales como PseAAC se puede lograr un buen rendimiento en la clasificación de péptidos antimicrobianos y sus diferentes funciones; pero, se debe tomar en cuenta que la selección de características que se realizó en este trabajo no fue robusta. Lo anterior invita a llevar a cabo un proceso de selección de características más elaborado

y analizar si el rendimiento de los clasificadores obtiene una mejora significativa.

La evaluación basada en validación por reserva de los clasificadores clásicos y profundos no asegura que todos los ejemplos sean considerados, al menos una vez, como elementos del conjunto de prueba, esto no permite medir la generalización de los modelos de forma robusta. Usar una validación cruzada de *k* pliegues podría resolver este problema, sin embargo, la división simple en *k* pliegues puede sesgar los resultados, por ello es común hacer un proceso repetido de esta validación (Kuncheva, 2014), lo que se traduce a mayores requerimientos de cómputo, sobre todo en lo modelos profundos. Lo anterior se convierte en una limitante debido a la infraestuctura con la que se cuenta atualmente. La carencia en la validación por reserva se puede ver compensada con la utilización del conjunto externo, el cual da una noción de cómo se comportarían los modelos en pruebas reales.

De las evaluaciones realizadas a los clasificadores clásicos y profundos (tablas 12, 13 y 14) no se aprecia un patrón que permita identificar la superioridad de alguno de los modelos, a excepción de la clase antibacteriana, donde en todas las evaluaciones el modelo profundo propuesto mantiene un mejor desempeño. Basándonos únicamente en la exactitud de los clasificadores no se podría establecer una medida de superioridad por parte de alguna arquitectura. Los resultados anteriores concuerdan con lo estípulado en el teorema de aproximación universal, el cual expone que cualquier función no lineal puede aproximarse con una red neuronal no profunda (así como cualquier modelo matemáticamente similar, como los modelos clásicos) (Cybenko, 1989).

Considerando lo anterior, se puede comparar el desempeño de los modelos clásicos y profundos en función del número de hiperparámetros ajustables, la necesidad de seleccionar descriptores, el tiempo de entrenamiento y predicción, el dominio de aplicabilidad y la interpretabilidad de los modelos.

El algoritmo RF tiene una cantidad pequeña de hiperparámetros, en este estudio se ajustó arbitrariamente el número de árboles y el resto se usaron con sus valores por omisión, sin embargo, se obtuvo buen desempeño en la mayoría de las actividades analizadas. Por su parte, los modelos profundos dependen de una cantidad mayor de hiperparámetros ajustables, algunos de ellos fueron optimizados en esta investigación, idenpendientemente para cada conjunto de datos, explorando un total de 512 modelos

para cada actividad biológica, lo cual podría significar una ventaja por parte de esta arquitectura; por lo anterior, la comparación realizada podría no ser justa, para ello se debió ejercer una optimización de los hiperparámetros del algoritmo RF.

La DDN propuesta mostró un buen desempeño para discriminar entre los casos negativos y positivos de cada conjunto de datos sin la necesidad de requerir un trabajo previo de selección de características. Contrario a lo anterior, para entrenar los modelos clásicos se llevó a cabo un proceso de selección de características para encontrar los descriptores moleculares con mayor información sobre la relación estructura-actividad en cada conjunto de datos, agregando con esto un esfuerzo extra para la construcción de los clasificadores. A pesar de esto, solo se exploraron 32 modelos de clasificación para cada actividad analizada, esta cantidad es considerablemente menor a la cantidad de modelos construidos para encontrar el óptimo de los clasificadores profundos.

Respecto al tiempo de entrenamiento, existe una notable diferencia entre los modelos clásicos y profundos, siendo los primeros los que requieren una menor inversión. Esta diferencia se ve reflejada principalmente en el proceso de validación por reserva, tomando en cuenta el conjunto antimicrobiano (conjunto de datos de mayor cardinalidad) el clasificador basado en RF se toma aproximadamente 30 minutos para finalizar la evaluación, mientras que el modelo profundo propuesto se toma aproximadamente 15 horas, esto refleja una clara ventaja en cuanto al uso de este recurso por parte de los modelos clásicos.

Para medir la fiabiliad de las predicciones dadas por los modelos de clasificación, en una modelación QSAR, es importante definir el dominio de aplicabilidad del clasificador (Mathea *et al.*, 2016), el cual está determinado principalmente por los descriptores moleculares y características del conjunto de datos. El dominio de aplicabilidad en un modelo que no depende de descriptores moleculares, como la DNN propuesta, no se podría determinar. Por otra parte, es importante tener una noción o interpretabilidad de las predicciones hechas por los clasificadores. Los modelos basados en RF proveen una forma más simple de interpretar las predicciones dotando de reglas lógicas basadas en la importancia de los descriptores moleculares, mientras que la interpretabilidad de las DNN se dificulta debido a la cantidad de parámetros aprendidos (Muratov *et al.*, 2020).

5.1.2.1. Comparación del rendimiento de los modelos propuestos con los métodos del estado del arte

Para comparar justamente los modelos de la literatura y los modelos propuestos en esta investigación se debe poner mayor énfasis en los conjuntos externos, debido a que con estos conjuntos se puede tener una idea más cercana de cómo se comportarían los clasificadores en pruebas reales. Además, porque debido a la separación realizada para obtener los conjuntos de entrenamiento y prueba, los conjunto de prueba estarían bien representados por nuestros conjuntos de entrenamiento, esperando un mejor desempeño de los modelos aquí propuestos.

Bajo la consideración anterior, de las tablas 19 y 20, se puede observar que los modelos propuestos en este trabajo son más robustos para identificar péptidos con las actividades estudiadas. Aunque algunos modelos de la literatura obtenienen una métrica superior en los conjuntos externos, se puede observar que su especificidad en los conjuntos de prueba es baja, en otras palabras, podríamos decir que estos modelos no son capaces de discriminar casos negativos. Para apoyar esta afirmación sería interesante agregar ejemplos negativos a los conjuntos de datos externos, para poder medir todas las métricas de desempeño de los clasificadores considerados en este estudio.

Por otra parte, en la Tabla 16, se puede apreciar que los modelos construidos para clasificar actividad antimicrobiana predicen de forma correcta en promedio 29 % y 15 % un mayor número de secuencias que los modelos del estado del arte, en los conjuntos de prueba y externos, respectivamente. Siendo esto una mejora considerable al rendimiento de clasificadores para este problema.

Una limitante para la comparación entre métodos de clasificación es el desconocimiento del dominio de aplicabilidad de los modelos reportados en la literatura, esto para tener fiabilidad en las predicciones que estos reportan. Lo ideal sería evaluar a todos los clasificadores usando un conjunto de datos que estuviera dentro del dominio de aplicabiliadad de todos los modelos. Una forma de solucionar este inconveniente puede encontrarse en las evaluaciones que se realizaron bajo el enfoque jerárquico, sin embargo, esto solo permite trabajar dentro del dominio de aplicabilidad independiente de cada clasificador.

5.1.3. Ejemplo de aplicación. Predicción de actividades antimicrobianas en un metagenoma de esponjas marinas del Parque Nacional Cabo Pulmo

Las predicciones obtenidas en los péptidos analizados muestran que estos presentan potencial terapeútico, ya que algunas de ellas tienen una predicción positiva en la mayoría de los modelos usados, como la secuencia con identificador número 30, esta sería una candidata prometedora para análisis computacionales adicionales así como experimentales. Por otra parte, es necesario medir la capacidad para discriminar secuencias sin potencial de tener alguna de las actividades analizadas, tal es el caso de la secuencia con identificador número 28, la cual fue predicha como negativa por todos los clasificadores, validar esta predicción experimentalmente mostraría la aplicabilidad de los modelos en el desarrollo de fármacos basados en péptidos antimicrobianos.

Una limitante para estudiar directamente las secuencia mencionadas anteriormente (así como el resto de estas) es la longitud que presentan, debido a que en promedio estas secuencias tienen una longitud de 162 aminoácidos, por lo que sintetizarlas para análisis experimentales sería muy costoso. Debido a esto, el siguiente paso sería la digestión in silico de estas secuencias, haciendo un proceso de ventaneo sobre las mismas, y el análisis de los péptidos resultantes de dicha digestión, prediciendo su potencial actividad antimicrobiana con los clasificadores propuestos para cada actividad biológica estudiada.

5.2. Conclusiones

Este trabajo de investigación abordó el problema de clasificación multiclase de péptidos antimicrobianos, se desarollaron modelos de aprendizaje clásicos y profundos y se comparó su desempeño en este problema. Para ello, se propusieron cinco conjuntos de datos, se determinó una metodología para separar el conjunto de entrenamiento del de prueba y mantener representatividad en el espacio químico de las secuencias. Se entrenaron clasificadores con el algoritmo RF para representar a los modelos clásicos y se propuso una DNN para comparar su desempeño. Se selecionaron los modelos mejor evaluados de cada arquitectura y se evaluaron bajo un proceso de validación

por reserva repetida, con un conjunto de prueba independiente y con conjuntos externos. Por otra parte, se predijo la potencial actividad antimicrobiana de péptidos extraídos de un metagenoma de esponjas marinas del parque nacional Cabo Pulmo para poner en práctica los clasificadores propuestos. A continuación, se presentan las conclusiones a las que se llegaron en este trabajo de investigación:

- Los conjuntos de datos para las actividades antibacteriana, antifúngica, antiparasitaria y antiviral presentan una diversidad de comparable a superior respecto a los conjuntos propuestos en la literatura, para estas clases, sin llevar a cabo un proceso de reducción de redundancia.
- La métrica de parentesco, bajo su definición actual, podría dar una idea equivocada de la calidad de los datos en cuanto al nivel de separabilidad de los conjuntos.
- Los clasificadores entrenados con descriptores de ProtDCal seleccionados para cada actividad no muestran un patrón de superioridad respecto a los entrenados con descriptores de PseAAC, bajo el proceso de selección seguido en esta investigación.
- Los modelos de clasificación profundos propuestos pueden discriminar con buena exactitud casos positivos y negativos, de las actividades estudiadas, sin considerar descriptores moleculares, basándose únicamente en la secuencia de aminoácidos.
- No se observa superioridad en el desempeño de los clasificadores profundos respecto a los clasificadores clásicos en los conjuntos de datos propuestos para cada actividad biológica.
- Los modelos de clasificación profundos requieren la optimización de un mayor número de hiperparámetros, por tanto demandan más recursos computacionales, tomando periodos de tiempo más largos para entrenamiento, además, las predicciones son difíciles de interpretar.
- Los clasificadores propuestos en este trabajo tienen un desempeño de comparable a superior respecto a los métodos del estado del arte para cada una de las actividades estudiadas.

- Las secuencias con número de identificador 30, 32 y 52 de Cabo Pulmo son candidatas prometedoras para su evaluación experimental.
- Las secuencias con número de identificador 27, 28, 38, 42, 51, 56 y 63 de Cabo Pulmo deben estudiarse para medir la capacidad de discriminar casos negativos de los modelos propuestos.

5.2.1. Trabajo futuro

Derivadas de las limitantes y los resultados del presente trabajo, se proponen a continuación ideas de investigación a explorar para extender los alcances del mismo.

Modelos de regresión

En la literatura se ha abordado la predicción de actividades biológicas de AMPs desde un enfoque de regresión, donde se predice una variable continua en lugar de una etiqueta o variable discreta, por ejemplo, la concentración mínima inhibitoria (MIC) necesaria para considerar que un péptido presenta alguna función deseada (Witten y Witten, 2019). Este enfoque permite dejar a un lado la construcción de un conjunto negativo, además, podría suponer una mayor usabilidad para los experimentalistas. Por ello se propone la construcción de un modelo de regresión, obteniendo los MICs de cada secuencia en los conjuntos de datos propuestos, disponibles en el trabajo de Aguilera-Mendoza *et al.* (2019).

Redes Neuronales Convolucionales de Grafos

Recientemente Stokes *et al.* (2020) mostraron que las redes neuronales convolucionales de grafos son capaces de identificar moléculas pequeñas con potencial uso terapeútico, lo anterior motiva a probar esta arquitectura de red en la identificación de AMPs y sus funciones específicas, aplicando la arquitectura propuesta por Yang *et al.* (2019).

Aprendizaje por refuerzo para el diseño de péptidos

El aprendizaje por refuerzo ha sido usado para el diseño de nuevas moléculas pequeñas (Vamathevan *et al.*, 2019), por esto se propone usar aprendizaje por refuer-

zo para el diseño de nuevos péptidos. Para ello, se propone hacer una reducción de los aminoácidos basada en las similitudes de los vectores generados por la capa de encrustamiento de la red neuronal propuesta, esto con la intención de reducir el espacio de búsqueda. Además, usar transferencia de conocimiento (Jia *et al.*, 2018) de los modelos entrenados en esta investigación para construir un modelo generador de secuencias.

Evaluación experimental de las secuencias predichas con potencial uso terapeútico

Se propone realizar un procesamiento de las secuencias predichas con potencial uso terapeútico para reducir la longitud de estas, basado en un enfoque de ventaneo para obtener subsecuencias y predicir su posible actividad antimicrobiana. Lo anterior es con la intención de conseguir péptidos candidatos con las actividades estudiadas y que se puedan validar de manera experimental, así como aquellos que muestren carecer de estas, es decir, aquellos que son predichos como negativos.

Literatura citada

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., y Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. En: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. pp. 265–283.

Agrawal, P., Bhalla, S., Chaudhary, K., Kumar, R., Sharma, M., y Raghava, G. P. (2018). In silico approach for prediction of antifungal peptides. *Frontiers in microbiology*, **9**: 323.

Aguilera-Mendoza, L., Marrero-Ponce, Y., Tellez-Ibarra, R., Llorente-Quesada, M. T., Salgado, J., Barigye, S. J., y Liu, J. (2015). Overlap and diversity in antimicrobial peptide databases: compiling a non-redundant set of sequences. *Bioinformatics*, **31**(15): 2553–2559.

Aguilera-Mendoza, L., Marrero-Ponce, Y., Beltran, J. A., Tellez Ibarra, R., Guillen-Ramirez, H. A., y Brizuela, C. A. (2019). Graph-based data integration from bioactive peptide databases of pharmaceutical interest: toward an organized collection enabling visual network analysis. *Bioinformatics*, **35**(22): 4739–4747.

Aly, M. (2005). Survey on multiclass classification methods. *Neural Netw*, **19**: 1–9.

Aoki, W. y Ueda, M. (2013). Characterization of antimicrobial peptides toward the development of novel antibiotics. *Pharmaceuticals*, **6**(8): 1055–1081.

Bahar, A. A. y Ren, D. (2013). Antimicrobial peptides. *Pharmaceuticals*, **6**(12): 1543–1575.

Bahdanau, D., Cho, K., y Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Beltran, J. A., Del Rio, G., y Brizuela, C. A. (2020). An automatic representation of peptides for effective antimicrobial activity classification. *Computational and structural biotechnology journal*, **18**: 455–463.

Bengio, Y., Simard, P., y Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, **5**(2): 157–166.

Bhadra, P., Yan, J., Li, J., Fong, S., y Siu, S. W. (2018). Ampep: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Scientific reports*, **8**(1): 1–10.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bonaccorso, G. (2020). *Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work*. Packt Publishing Ltd.

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

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, **78**(1): 1–3.

Brogden, K. A. (2005). Antimicrobial peptides: pore formers or metabolic inhibitors in bacteria? *Nature reviews microbiology*, **3**(3): 238–250.

Bull, A. T. y Stach, J. E. (2007). Marine actinobacteria: new opportunities for natural product search and discovery. *Trends in microbiology*, **15**(11): 491–499.

Burkov, A. (2019). *The hundred-page machine learning book*, Vol. 1. Autor. Quebec City, Can.

Chai, T. y Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, **7**(3): 1247–1250.

Chandrasekaran, B., Abed, S. N., Al-Attraqchi, O., Kuche, K., y Tekade, R. K. (2018). Computer-aided prediction of pharmacokinetic (admet) properties. En: *Dosage Form Design Parameters*. Elsevier, pp. 731–755.

Chandrashekar, G. y Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, **40**(1): 16–28.

Chen, Z., Zhao, P., Li, F., Marquez-Lago, T. T., Leier, A., Revote, J., Zhu, Y., Powell, D. R., Akutsu, T., Webb, G. I., *et al.* (2020). ilearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of dna, rna and protein sequence data. *Briefings in bioinformatics*, **21**(3): 1047–1057.

Cho, K., Van Merriënboer, B., Bahdanau, D., y Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Chollet, F. *et al.* (2015). Keras. Recuperado: marzo de 2020, de: `https://keras.io`.

Chou, K.-C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, **43**(3): 246–255.

Chung, C.-R., Kuo, T.-R., Wu, L.-C., Lee, T.-Y., y Horng, J.-T. (2019). Characterization and identification of antimicrobial peptides with different functional activities. *Briefings in bioinformatics*. `https://doi.org/10.1093/bib/bbz043`.

Chung, C.-R., Jhong, J.-H., Wang, Z., Chen, S., Wan, Y., Horng, J.-T., y Lee, T.-Y. (2020). Characterization and identification of natural antimicrobial peptides on different organisms. *International Journal of Molecular Sciences*, **21**(3): 986.

Chung, J., Gulcehre, C., Cho, K., y Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, **2**(4): 303–314.

Dempster, A. P., Laird, N. M., y Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1): 1–22.

Dharmaraj, S. (2010). Marine streptomyces as a novel source of bioactive substances. *World Journal of Microbiology and Biotechnology*, **26**(12): 2123–2139.

Donalek, C. (2011). Supervised and unsupervised learning. *Astronomy Colloquia. USA*. Recuperado: agosto de 2020, de: `https://sites.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf`.

Esteves, A. I., Hardoim, C. C., Xavier, J. R., Gonçalves, J. M., y Costa, R. (2013). Molecular richness and biotechnological potential of bacteria cultured from irciniidae sponges in the north-east atlantic. *FEMS microbiology ecology*, **85**(3): 519–536.

Fan, L., Sun, J., Zhou, M., Zhou, J., Lao, X., Zheng, H., y Xu, H. (2016). Dramp: a comprehensive data repository of antimicrobial peptides. *Scientific reports*, **6**(1): 1–7.

Fjell, C. D., Hiss, J. A., Hancock, R. E., y Schneider, G. (2012). Designing antimicrobial peptides: form follows function. *Nature reviews Drug discovery*, **11**(1): 37–51.

Frank, E., Hall, M., y Witten, I. (2016). The weka workbench. online appendix for "data mining: Practical machine learning tools and techniques".

Gabere, M. N. y Noble, W. S. (2017). Empirical comparison of web-based antimicrobial peptide prediction tools. *Bioinformatics*, **33**(13): 1921–1929.

Goodarzi, M., Dejaegher, B., y Heyden, Y. V. (2012). Feature selection methods in QSAR studies. *Journal of AOAC International*, **95**(3): 636–651.

Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press. Recuperado: agosto de 2020, de: `http://www.deeplearningbook.org`.

Graves, A., Mohamed, A.-r., y Hinton, G. (2013). Speech recognition with deep recurrent neural networks. En: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, pp. 6645–6649.

Guerrero-Vázquez, K. (2019). *Aprendizaje de máquina para la identificación de péptidos inductores de autofagia.*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. 99 pp.

Gulder, T. A. y Moore, B. S. (2009). Chasing the treasures of the sea—bacterial marine natural products. *Current opinion in microbiology*, **12**(3): 252–260.

Gull, S., Shamim, N., y Minhas, F. (2019). Amap: Hierarchical multi-label prediction of biologically active and antimicrobial peptides. *Computers in biology and medicine*, **107**: 172–181.

Guyon, I. y Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, **3**(Mar): 1157–1182.

Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., Couger, M. B., Eccles, D., Li, B., Lieber, M., *et al.* (2013). De novo transcript sequence reconstruction from rna-seq using the trinity platform for reference generation and analysis. *Nature protocols*, **8**(8): 1494–1512.

Hamid, M.-N. y Friedberg, I. (2018). Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *Bioinformatics*, **35**(12): 2009–2016.

Hamid, M.-N. y Friedberg, I. (2019). Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *Bioinformatics*, **35**(12): 2009–2016.

Henikoff, S. y Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, **89**(22): 10915–10919.

Hochreiter, S. y Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8): 1735–1780.

Huang, Y., Niu, B., Gao, Y., Fu, L., y Li, W. (2010). Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, **26**(5): 680–682.

Jhong, J.-H., Chi, Y.-H., Li, W.-C., Lin, T.-H., Huang, K.-Y., y Lee, T.-Y. (2018). dbamp: an integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data. *Nucleic acids research*, **47**(D1): D285–D297.

Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Moreno, I. L., Wu, Y., *et al.* (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. En: *Advances in neural information processing systems*. pp. 4480–4490.

Jiménez, L. F. y Merchant, H. (2003). *Biología celular y molecular*. Pearson educación México.

Jin, X. y Han, J. (2010). *Expectation Maximization Clustering. En: Encyclopedia of Machine Learning*, pp. 382–383. Springer, Boston, MA.

Joseph, S., Karnik, S., Nilawe, P., Jayaraman, V. K., y Idicula-Thomas, S. (2012). Classamp: a prediction tool for classification of antimicrobial peptides. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **9**(5): 1535–1538.

Kohavi, R., John, G. H., *et al.* (1997). Wrappers for feature subset selection. *Artificial intelligence*, **97**(1-2): 273–324.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. En: *Advances in neural information processing systems*. pp. 1097–1105.

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Lago-Lestón, M., Hardoim, C., Cox, C., Pires, F., Goncalves, J., CJ, Xavier, J., y Costa, R. (2013). Tackling the specificity of the marine sponge microbiome: a biogeographical approach. *Microbial diversity 2013: microbial interactions in complex ecosystems*, pp. 143–146.

Lee, H.-T., Lee, C.-C., Yang, J.-R., Lai, J. Z., y Chang, K. Y. (2015). A large-scale structural classification of antimicrobial peptides. *BioMed research international*, **2015**. `https://doi.org/10.1155/2015/475062`.

Lewies, A., Du Plessis, L. H., y Wentzel, J. F. (2018). Antimicrobial peptides: the achilles' heel of antibiotic resistance? *Probiotics and antimicrobial proteins*, pp. 1–12.

Lewies, A., Du Plessis, L. H., y Wentzel, J. F. (2019). Antimicrobial peptides: the achilles' heel of antibiotic resistance? *Probiotics and antimicrobial proteins*, **11**(2): 370–381.

Li, Y., Xiang, Q., Zhang, Q., Huang, Y., y Su, Z. (2012). Overview on the recent study of antimicrobial peptides: origins, functions, relative mechanisms and application. *Peptides*, **37**(2): 207–215.

Lin, W. y Xu, D. (2016). Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types. *Bioinformatics*, **32**(24): 3745–3752.

Lin, Y., Cai, Y., Liu, J., Lin, C., y Liu, X. (2019). An advanced approach to identify antimicrobial peptides and their function types for penaeus through machine learning strategies. *BMC bioinformatics*, **20**(8): 291.

Lorena, A. C., De Carvalho, A. C., y Gama, J. M. (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, **30**(1-4): 19.

Mathea, M., Klingspohn, W., y Baumann, K. (2016). Chemoinformatic classification methods and their applicability domain. *Molecular Informatics*, **35**(5): 160–180.

Matsuzaki, K. (2019). *Antimicrobial Peptides: Basics for Clinical Application*, Vol. 1117. Springer.

McKee, T. y McKee, J. R. (2009). Bioquímica: las bases moleculares de la vida. Reporte técnico. (No. Sirsi) i9789701070215.

Meher, P. K., Sahu, T. K., Saini, V., y Rao, A. R. (2017). Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into chou's general pseaac. *Scientific reports*, **7**(1): 1–12.

Meléndrez-Carballo, G. (2018). *Análisis de transcriptomas para el descubrimiento de péptidos antimicrobianos*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California. 113 pp.

Midura-Nowaczek, K. y Markowska, A. (2014). Antimicrobial peptides and their analogs: searching for new potential therapeutics. *Perspectives in medicinal chemistry*, **6**: PMC–S13215.

Minkiewicz, P., Iwaniak, A., y Darewicz, M. (2019). Biopep-uwm database of bioactive peptides: Current opportunities. *International Journal of Molecular Sciences*, **20**(23): 5978.

Mitchel, T. (1997). *Machine learning*, Vol. 1. McGraw-Hill.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Molina, L. C., Belanche, L., y Nebot, À. (2002). Feature selection algorithms: A survey and experimental evaluation. En: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*. IEEE, pp. 306–313.

Muratov, E. N., Bajorath, J., Sheridan, R. P., Tetko, I. V., Filimonov, D., Poroikov, V., Oprea, T. I., Baskin, I. I., Varnek, A., Roitberg, A., Isayev, O., Curtalolo, S., Fourches, D., Cohen, Y., Aspuru-Guzik, A., Winkler, D. A., Agrafiotis, D., Cherkasov, A., y Tropsha, A. (2020). QSAR without borders. *Chem. Soc. Rev.*, **49**: 3525–3564.

Needleman, S. B. y 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., A., Katanforoosh, K., y Bensouda-Mourri, Y. (s.f.). Gated recurrent unit (GRU). En: Sequences models. Recuperado: marzo de 2020, de: `https://www.coursera.org/learn/nlp-sequence-models?`

Peters, B. M., Shirtliff, M. E., y Jabra-Rizk, M. A. (2010). Antimicrobial peptides: primeval molecules or future drugs? *PLoS Pathog*, **6**(10): e1001067.

Pirtskhalava, M., Gabrielian, A., Cruz, P., Griggs, H. L., Squires, R. B., Hurt, D. E., Grigolava, M., Chubinidze, M., Gogoladze, G., Vishnepolsky, B., *et al.* (2015). Dbaasp v. 2: an enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides. *Nucleic acids research*, **44**(D1): D1104–D1112.

Rahman, H., Austin, B., Mitchell, W. J., Morris, P. C., Jamieson, D. J., Adams, D. R., Spragg, A. M., y Schweizer, M. (2010). Novel anti-infective compounds from marine bacteria. *Marine drugs*, **8**(3): 498–518.

Roca, I., Akova, M., Baquero, F., Carlet, J., Cavaleri, M., Coenen, S., Cohen, J., Findlay, D., Gyssens, I., Heure, O., *et al.* (2015). The global threat of antimicrobial resistance: science for intervention. *New microbes and new infections*, **6**: 22–29.

Romero-Molina, S., Ruiz-Blanco, Y. B., Green, J. R., y Sanchez-Garcia, E. (2019). Protdcal-suite: A web server for the numerical codification and functional analysis of proteins. *Protein Science*, **28**(9): 1734–1743.

Roy, K., Kar, S., y Das, R. N. (2015). *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, **323**(6088): 533–536.

Russell, S. y Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Pearson.

Shannon, C. E. y Weaver, W. (1949). The mathematical theory of communication. university of illinois. *Urbana*, **117**.

Shrestha, A. y Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, **7**: 53040–53065.

Singh, S. y Gupta, P. (2014). Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, **27**(27): 97–103.

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackerman, Z., *et al.* (2020). A deep learning approach to antibiotic discovery. *Cell*, **180**(4): 688–702.

Su, X., Xu, J., Yin, Y., Quan, X., y Zhang, H. (2019). Antimicrobial peptide identification using multi-scale convolutional network. *BMC bioinformatics*, **20**(1): 1–10.

Sung, W.-K. (2009). *Algorithms in bioinformatics: A practical introduction*. CRC Press.

Sutskever, I., Vinyals, O., y Le, Q. V. (2014). Sequence to sequence learning with neural networks. En: *Advances in neural information processing systems*. pp. 3104–3112.

Szalkai, B. y Grolmusz, V. (2018). Near perfect protein multi-label classification with deep neural networks. *Methods*, **132**: 50–56.

Teta, R., Gurgui, M., Helfrich, E. J., Künne, S., Schneider, A., Van Echten-Deckert, G., Mangoni, A., y Piel, J. (2010). Genome mining reveals trans-at polyketide synthase directed antibiotic biosynthesis in the bacterial phylum bacteroidetes. *ChemBioChem*, **11**(18): 2506–2512.

Thappeta, K. R., Vikhe, Y. S., Yong, A. M., Chan-Park, M. B., y Kline, K. A. (2020). Combined efficacy of an antimicrobial cationic peptide polymer with conventional antibiotics to combat multidrug-resistant pathogens. *ACS Infectious Diseases*, **6**(5): 1228–1237.

Todeschini, R. y Consonni, V. (2008). *Handbook of molecular descriptors*, Vol. 11. John Wiley & Sons.

Torres, M. D. T. y de la Fuente-Nunez, C. (2019). Toward computer-made artificial antibiotics. *Current opinion in microbiology*, **51**: 30–38.

Tsoumakas, G. y Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, **3**(3): 1–13.

UniProt, C. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, **47**(D1): D506–D515.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., *et al.* (2019). Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, **18**(6): 463–477.

Veltri, D., Kamath, U., y Shehu, A. (2018). Deep learning improves antimicrobial peptide recognition. *Bioinformatics*, **34**(16): 2740–2747.

Villoutreix, B. O., Renault, N., Lagorce, D., Sperandio, O., Montes, M., y Miteva, M. A. (2007). Free resources to assist structure-based virtual ligand screening experiments. *Current Protein and Peptide Science*, **8**(4): 381–411.

Voigt, J. H., Bienfait, B., Wang, S., y Nicklaus, M. C. (2001). Comparison of the nci open database with seven large chemical structural databases. *Journal of chemical information and computer sciences*, **41**(3): 702–712.

Waghu, F. H., Barai, R. S., Gurung, P., y Idicula-Thomas, S. (2016). Campr3: a database on sequences, structures and signatures of antimicrobial peptides. *Nucleic acids research*, **44**(D1): D1094–D1097.

Wang, G., Li, X., y Wang, Z. (2015). Apd3: the antimicrobial peptide database as a tool for research and education. *Nucleic acids research*, **44**(D1): D1087–D1093.

Wang, P., Ge, R., Liu, L., Xiao, X., Li, Y., y Cai, Y. (2017). Multi-label learning for predicting the activities of antimicrobial peptides. *Scientific reports*, **7**(1): 2202.

Wei, L., Zhou, C., Su, R., y Zou, Q. (2019). Pepred-suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics*, **35**(21): 4272–4280.

Witten, J. y Witten, Z. (2019). Deep learning regression model for antimicrobial peptide design. *BioRxiv*, p. 692681.

World Health Organization *et al.* (2014). *Antimicrobial resistance: global report on surveillance*. Autor. Recuperado: agosto de 2020, de: `https://apps.who.int/iris/handle/10665/112642`.

Xiao, X., Wang, P., Lin, W.-Z., Jia, J.-H., y Chou, K.-C. (2013). iamp-2l: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Analytical biochemistry*, **436**(2): 168–177.

Yan, J., Bhadra, P., Li, A., Sethiya, P., Qin, L., Tai, H. K., Wong, K. H., y Siu, S. W. (2020). Deep-ampep30: Improve short antimicrobial peptides prediction with deep learning. *Molecular Therapy-Nucleic Acids*, **20**: 882 – 894.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., *et al.* (2019). Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, **59**(8): 3370–3388.

Young, S. S., Yuan, F., y Zhu, M. (2012). Chemical descriptors are more important than learning algorithms for modelling. *Molecular informatics*, **31**(10): 707–710.

Zasloff, M. (2002). Antimicrobial peptides of multicellular organisms. *nature*, **415**(6870): 389–395.

Anexo A. Resultados del proceso de selección de características y descriptores moleculares de ProtDCal seleccionados.

Tabla 29. Número de descriptores seleccionados trás primer proceso de selección de características.

Actividad	Representación	Total de descriptores	No. de Seleccionados
Antibacteriana	NO	6181	334
	KH	12835	1042
	ES	12835	1763
	AC1	12835	728
	AC2	12835	728
	AC3	12835	706
	AC4	12835	675
	AC5	12835	632
Antifúngica	NO	6181	718
	KH	12835	2212
	ES	12835	3225
	AC1	12835	1400
	AC2	12835	1431
	AC3	12835	1429
	AC4	12835	1423
	AC5	12835	1374
Antiparasitaria	NO	6181	683
	KH	12835	1319
	ES	12835	2329
	AC1	12835	751
	AC2	12835	746
	AC3	12835	746
	AC4	12835	707
	AC5	12835	682
Antiviral	NO	6181	464
	KH	12835	1169
	ES	12835	1723
	AC1	12835	642
	AC2	12835	636
	AC3	12835	599
	AC4	12835	576
	AC5	12835	549

Este proceso de selección se basó en el algoritmo *FS_ENTROPY_CORRELATION*

Tabla 30. Número de descriptores seleccionados trás segundo proceso de selección de características.

Actividad	Representación	Total de descriptores	No. de Seleccionados
Antibacteriana	NO	6181	334
	KH	12835	1042
	ES	12835	1763
	AC2_S	1456	1371
	AC5_S	3469	3200
	ALL	6608	6040
Antifúngica	NO	6181	718
	KH	12835	2212
	ES	12835	3225
	AC2_S	2831	2691
	AC5_S	7057	6593
	ALL	13212	12248
Antiparasitaria	NO	6181	683
	KH	12835	1319
	ES	12835	2329
	AC2_S	1497	1389
	AC5_S	3632	3305
	ALL	7963	7207
Antiviral	NO	6181	464
	KH	12835	1169
	ES	12835	1723
	AC2_S	1278	1225
	AC5_S	3002	2840
	ALL	6358	5904

Este proceso de selección se basó en el algoritmo *FS_ENTROPY_CORRELATION*.

Tabla 31. Número de descriptores seleccionados de la unión de las representaciones usadas para formar el modelo por ensamble por actividad.

Actividad	No. de descriptores	No. de Seleccionados	Nomenclatura
Antibacteriana	108	61	WR_EM_61
Antifúngica	50	21	WR_EM_21
Antiparasitaria	117	43	WR_EM_43
Antiviral	119	69	WR_EM_69

Tabla 32. Descriptores moleculares de ProtDCal utilizados para entrenar el modelo de actividad antimicrobiana.

Representación	Descriptores moleculares seleccionados
WR_AMP_131	Gw(U)_KH3_NPR_N1, DHf_AC4_NPR_N2, Ap_ES_ALR_N3, Pb_AC3_PLR_N1, Ap_KH3_PLR_N2, ISA_AC4_NPR_N3, Pb_AC5_NPR_N1, Xi_NO_PLR_N1, Xi_NO_PLR_P2, ISA_NO_NPR_N3, ISA_AC3_ALR_N2, Mw_AC2_ALR_N3, Mw_ES_NPR_RA, ISA_AC2_ALR_N3, ISA_AC5_NPR_P2, DHf_AC5_PLR_N3, ISA_AC2_AHR_N3, ISA_ES_AHR_I50, Gs(U)_NO_PLR_Ar, Xi_ES_NPR_RA, ISA_AC1_AHR_N2, W(U)_ES_PLR_N2, DHf_KH3_ALR_DE, Xi_ES_ALR_RA, Mw_AC4_PLR_N3, Mw_ES_ALR_N2, Mw_AC3_NPR_N3, Xi_NO_NPR_DE, Pt_NO_BSR_N1, Gs(U)_ES_NPR_RA, ISA_KH3_PLR_N2, Gs(U)_AC3_ALR_N3, Xi_ES_ALR_DE, Xi_NO_NPR_N1, W(U)_NO_ALR_P2, Gs(U)_AC2_NPR_N3, Gs(U)_ES_PLR_RA, ISA_ES_PLR_RA, Ap_AC3_AHR_DE, ISA_NO_ALR_G, Gs(U)_NO_PLR_N2, HP_NO_ALR_N2, Gs(U)_KH3_ALR_N3, Z2_KH3_AHR_N2, Xi_NO_NPR_Ar, ISA_ES_ALR_I50, Ap_KH3_AHR_RA, Z1_AC5_AHR_N3, Ap_NO_PLR_DE, ISA_AC4_ALR_N3, Gs(U)_ES_ALR_Q2, HP_NO_NPR_G, Z3_AC1_PLR_N2, Ap_NO_PLR_M, L1-9_NO_NPR_V, W(U)_NO_ALR_N2, Gs(U)_ES_NPR_G, Z2_NO_PLR_N1, Gs(U)_AC1_NPR_N2, Xi_NO_PLR_N2, W(U)_ES_ALR_N2, Xi_AC3_AHR_P3, Z3_NO_NPR_N2, ISA_AC4_NPR_P3, Gs(U)_ES_NPR_N2, ISA_KH3_AHR_N3, Xi_AC1_AHR_N2, ISA_ES_PLR_N2, Xi_KH3_AHR_N3, ISA_NO_PLR_N2, Xi_NO_ALR_N2, IP_NO_AHR_N1, Gs(U)_AC2_NPR_N2, ISA_AC1_NPR_N3, W(U)_NO_RTR_N2, IP_AC5_PLR_N2, Gs(U)_AC2_PLR_N3, Xi_AC2_AHR_N3, Gs(U)_ES_ALR_I50, DHf_ES_PLR_N3, ECI_NO_PLR_N2, L1-9_NO_PLR_N2, ISA_AC3_NPR_N3, DHf_AC1_AHR_N1, DHf_NO_AHR_N2, ISA_KH3_ALR_RA, ISA_AC2_PLR_N3, ISA_NO_AHR_N2, HP_NO_AHR_N1, ISA_AC2_NPR_N3, Mw_AC1_NPR_N1, ISA_ES_NPR_Q1, Gs(U)_NO_NPR_Ar, DHf_KH3_AHR_DE, Gs(U)_AC1_NPR_N3, ISA_AC5_AHR_N3, Gw(U)_KH3_AHR_N3, DHf_KH3_PLR_DE, Ap_AC4_AHR_P3, ISA_AC5_NPR_N2, Z1_ES_PLR_RA, Gs(U)_AC5_NPR_N2, ISA_KH3_PLR_RA, Gs(U)_ES_AHR_N2, Z3_NO_PLR_N2, Xi_AC4_NPR_N3, ISA_AC4_PLR_N2, W(U)_AC2_ALR_N2, ISA_AC3_AHR_N3, IP_AC1_PLR_N2, Xi_KH3_ALR_RA, ISA_AC2_AHR_N2, W(U)_NO_BSR_N2, Z2_NO_NPR_N2, Gs(U)_NO_PLR_N1, DHf_ES_PLR_RA, ISA_AC5_AHR_P2, Ap_ES_NPR_N3, Gs(U)_NO_NPR_G, ISA_NO_NPR_DE, DHf_NO_PLR_N2, Ap_AC1_ALR_N2, ISA_AC1_ALR_N3, Ap_AC3_NPR_N1, Gs(U)_ES_ALR_RA, Z3_NO_NPR_N1, Xi_AC2_NPR_N3, Gs(U)_ES_BSR_G, ISA_NO_NPR_Ar, Xi_ES_PLR_N2, Ap_NO_AHR_N1

Tabla 33. Descriptores moleculares de ProtDCal utilizados para entrenar los modelos representativos de cada actividad (antibacteriana, antifúngica, antiparasitaria y antiviral).

Representación	Descriptores moleculares seleccionados
WR_EM_61	Mw_AC2_ALR_N3, ISA_NO_AHR_N2, ISA_ES_NPR_RA, Gs(U)_AC3_NPR_N3, Gs(U)_ES_NPR_RA, Gs(U)_ES_BSR_G, Gs(U)_AC2_AHR_N2, Mw_AC2_NPR_N2, Gs(U)_NO_PLR_DE, ISA_AC2_AHR_N2, Xi_AC2_PLR_N3, Gs(U)_AC5_NPR_N2, Gs(U)_NO_AHR_Ar, Gs(U)_ES_NPR_I50, DHf_NO_AHR_N2, ISA_ES_NPR_I50, Xi_AC1_NPR_N3, Gs(U)_ES_PLR_RA, Xi_NO_NPR_Ar, Gs(U)_NO_NPR_N2, IP_AC1_PLR_N2, W(U)_ES_PLR_N2, ISA_AC1_ALR_N3, ISA_ES_AHR_I50, IP_AC5_PLR_N2, Xi_ES_ALR_RA, ISA_ES_PLR_N2, Gs(U)_NO_AHR_N3, DHf_ES_PLR_N3, Ap_AC2_AHR_N3, Gs(U)_AC1_NPR_N3, ISA_ES_NPR_N2, DHf_ES_PLR_RA, ISA_AC4_NPR_N3, ISA_ES_AHR_N2, DHf_AC4_NPR_N2, Z2_NO_NPR_N1, Gs(U)_NO_PLR_P2, Mw_ES_NPR_RA, ISA_AC2_ALR_N3, ISA_AC2_NPR_N3, DHf_AC4_AHR_N3, Gs(U)_AC2_PLR_N3, Xi_AC2_AHR_N2, W(U)_AC2_ALR_N2, Ap_NO_PLR_M, ISA_NO_PLR_N2, Ap_AC2_NPR_N1, Ap_AC2_PLR_N2, Mw_ES_ALR_N2, Ap_ES_AHR_N2, ISA_AC1_AHR_N3, Ap_NO_NPR_N2, Gs(U)_NO_PLR_Ar, Gs(U)_ES_AHR_N2, Xi_ES_PLR_N2, Gs(U)_AC1_NPR_N2, Gs(U)_AC2_NPR_N2, ISA_AC3_AHR_N3, Xi_NO_PLR_N2, Gs(U)_AC1_AHR_N3
WR_EM_21	Mw_AC1_NPR_N1, Pb_AC3_PLR_N1, Ap_NO_AHR_N1, Z3_NO_AHR_N2, Pt_NO_BSR_N1, Gs(U)_ES_NPR_N1, ISA_NO_PLR_N2, IP_NO_PLR_N2, Ap_AC1_NPR_N1, Gs(U)_NO_PLR_N1, Gs(U)_NO_NPR_N1, Pt_NO_NPR_N1, DHf_AC1_AHR_N1, Xi_NO_PLR_N1, IP_NO_AHR_N1, Gw(U)_NO_PLR_N2, W(U)_AC2_NPR_N1, Xi_NO_AHR_N2, Xi_AC2_NPR_N3, Z1_ES_PLR_N1, DHf_AC4_AHR_N1
WR_EM_43	W(U)_NO_RTR_N2, ISA_AC5_AHR_P2, Gs(U)_AC4_AHR_P3, Z1_NO_NPR_G, Xi_AC3_ALR_N2, Z2_NO_NPR_N2, Ap_NO_AHR_N2, Xi_AC2_ALR_N3, Xi_NO_ALR_N2, Gs(U)_ES_NPR_I50, W(U)_NO_AHR_G, L1-9_NO_NPR_V, W(U)_NO_PLR_N3, ISA_AC3_NPR_N3, Ap_AC5_PLR_G, Xi_AC4_AHR_P2, Xi_AC4_AHR_N2, Gw(U)_NO_PLR_Ar, Gs(U)_NO_NPR_Ar, W(U)_ES_ALR_N2, ISA_ES_AHR_I50, Ap_AC4_AHR_P3, Gs(U)_AC4_NPR_P2, Xi_AC4_NPR_N2, ISA_AC1_AHR_N2, Xi_AC1_NPR_N2, ISA_AC5_AHR_N3, ISA_AC2_PLR_N3, HP_ES_ALR_I50, ISA_ES_NPR_Q1, Z2_NO_NPR_N1, Z2_ES_PLR_I50, ISA_ES_ALR_Q1, Gw(U)_NO_PLR_G, W(U)_NO_ALR_G, Pb_NO_ALR_N3, Gs(U)_AC1_AHR_P3, Z3_NO_NPR_N2, Gs(U)_AC3_ALR_N3, Z1_NO_ALR_G, Gs(U)_AC1_NPR_N2, Xi_AC2_NPR_N3, L1-9_NO_PLR_N2
WR_EM_69	Gs(U)_ES_PLR_MX, Gs(U)_NO_PLR_G, ISA_ES_NPR_RA, Gs(U)_ES_NPR_RA, Mw_ES_PLR_RA, Gs(U)_ES_BSR_G, Gs(U)_ES_NPR_Q2, Ap_NO_PLR_DE, Xi_NO_NPR_N2, Xi_KH3_ALR_RA, ISA_ES_PLR_RA, ISA_ES_PLR_P2, Xi_KH3_NPR_RA, Gs(U)_ES_ALR_G, Gs(U)_KH3_PLR_RA, ISA_ES_NPR_I50, ISA_AC3_NPR_N3, Xi_ES_NPR_DE, Gs(U)_ES_PLR_RA, Gs(U)_NO_PLR_N2, DHf_ES_NPR_RA, Xi_ES_NPR_RA, Ap_KH3_AHR_N3, ISA_AC4_AHR_N3, Gs(U)_NO_NPR_N2, Gs(U)_KH3_ALR_N3, Ap_ES_NPR_N3, Mw_NO_PLR_N2, Gs(U)_NO_NPR_Ar, Xi_AC2_AHR_N3, ISA_ES_AHR_I50, Mw_AC5_AHR_G, ISA_KH3_NPR_N3, ISA_AC4_AHR_P3, ISA_ES_ALR_G, Gw(U)_ES_AHR_G, DHf_ES_AHR_RA, Xi_NO_NPR_DE, ISA_AC4_ALR_N3, ISA_NO_NPR_N2, ISA_KH3_PLR_RA, Ap_ES_PLR_RA, DHf_ES_PLR_RA, ISA_AC4_NPR_N3, ISA_NO_PLR_N3, Z3_KH3_PLR_N2, Gs(U)_ES_NPR_Q1, Gs(U)_KH3_NPR_RA, Mw_ES_NPR_RA, Mw_AC5_PLR_Ar, Gw(U)_NO_PLR_G, ISA_KH3_ALR_RA, Xi_NO_NPR_N3, Xi_AC3_AHR_N3, Z3_KH3_AHR_N2, DHf_KH3_PLR_DE, Ap_NO_PLR_M, ISA_AC4_NPR_P3, Gs(U)_NO_NPR_G, ISA_ES_ALR_I50, ISA_KH3_NPR_RA, ISA_ES_PLR_N3, ISA_AC3_ALR_N3, ISA_AC3_AHR_N3, ISA_ES_AHR_Q2, DHf_KH3_NPR_DE, Xi_NO_PLR_N2, Gs(U)_ES_ALR_Q2, Z2_KH3_AHR_N2

Anexo B. Secuencias de un metagenoma de esponja marina consideradas para predecir su potencial actividad antimicrobiana.

Tabla 34. Secuencias consideradas para predecir péptidos con potencial uso terapeútico, obtenidas de un metagenoma de esponjas marinas del Parque Nacional Cabo Pulmo.

Secuencias

>Sequence_1: AAAAAADVPVVRGIDHAVSAAEAEDFLDGLGPGRAMMLKAVAGGGGRGSRMVERADDVAAVYERCRAEAEAAFGNGELYVEEFMRRARHVEVQILGDLHGAVAHLGERECSVQRHFQKIVEVAPAPGLDDAVRDALVDAAVRFAASVGYSNAGTFEFLVDVSGEGGGAASRERGAFAFIETNARLQVEHTV

>Sequence_2: AAAAADGGTAVAREVDIRSGESIEALFSGLESQGFRPQGLVSTPGINVRKRLADTDEEEFERVVSLNLKGTFHALRHGAASLARSGGGSIVLLSSIRSQVVEPGQGVYAATKAGIVQLARTAAAEWAASGVRVNALAPGVVETPLTAPIENRPAWKQAYTDKTLLGRWAAAEE

>Sequence_3: AAAAAGKIIACEKPLAMDAAEGQRLVDAVAASGKPNMVWFNYRRVPAIALAKQIVDEGRIGRVFHYRSRYLQDWTISADVPLGGNTLWRLDKSVSGSGVTGDLLAHSIDTA

>Sequence_4: AAAAAIGVLNSRMMEIRTVTVSGSDRVSADSIAQLAAVTGENILLADLDAARARIREQPLIRDASLRREWPNTIHVHVWERTPWVRWVAQGEVWAVDREGVVLDGVEAPADSIIVRQVSSLPSIRAGTHVGLDAVALIQRIEQVGVPRDGPDIVAFEWSLRQGLTVVTRHGRVTFGDAQGFEFKYLVWQELEFEAQRRGEPLLIADLRFGTRPAVEIGLGLGRATRITPEGVVGGDPS

>Sequence_5: AAAAALDFETAAKYRDTLKNVQQAITTQSLDTVSAADEDVIGIAARTDIACVQLLRVRDGKLLEREHYYLNDADPKSLATALGAFISQYYQNAVFVPKTIVLPMSIEGIELIEAWLSEKRGNRVALHVPRAGRLRKLQALASKNAEVLLTQREQSVVYSSGIDPALVELQELIGLAQPLRR IEAFDISNLGDRFAVGS

>Sequence_6: AAAAAPVAPVVVVDELEVLGERPVPRFEVGVVEARAAVQQEQRRALAHAGAVAHQLGAVGIEEEAHVADVDVHDVHPCAAAERDSLHPQANRIKLPPGQGAPGRASTRRARRSFRSRLPR

>Sequence_7: AAAAAQVQHYRDDRNGQTEGTKPPAVLVIQGVAERHRLRAGGDAEQDQHDRRPAAGNCCNTGHDPEPDAAPHLGRRRRATAVGVRSAHRSDASSSATTGAWSLAIMPSCFLR

>Sequence_8: AAAACMSPHHFQRTFRRWVGISPTRFLQYLTLERAKASLAENASVLEASWDAGLSGPGRLHDLFVTHEAVTPGDYKRRGAGLAIRYGIHESPFGRCILMATDRGLCGLAFAAEGQDRAGLENMRRRWPAAAFME

>Sequence_9: AAAADGFASTRALPSHRRSEILQNMGGLLRERFDEIVDAMILEGGKNRKTAIGETTRALETLKISAEEARRIGGEVFSLDWTAAGANRQGFTRRQPIGTVLGITPFNYPVNLACHKIGPAIAAGNPIIIKPAEKTPLSSVILAEIVLEAGFPPQAF

>Sequence_10: AAAADLTDQNLQSRIRGVLLMALSNRFGWLVLTTGNKTETSVGYTTLYGDTAGAYAVIRDVPKLLVYELCEWRNRRGDGERIPRACITKPPSAELRPDQRDDQS

>Sequence_11: AAAADPPPVSSLSLERGCLADEPAPALPRYRLDGEQRDAIRSFLHRQQQFPDVSAAPVYDFYRRISRLRCTSCHVMDHIGGGEGGGAPVLTGAGEKLRPDYLEQVLTEGVRLRTWLSQRMPRFPRDQVEPLIQGFAKASGLDPSAPAVESVTSQRETRGGLILLGQDTEGKSLGCIGCHDWGEFQARGEKAPQLAEAARRLRYPWYRRFMLNPARILSGTSM

>Sequence_12: AAAAEADQAGLLLDRAELAERADVPAFVVDLVIDAGLLAPLDNGSEGELFGSDDVDVLAAARTLVGEGVAIEELAALAMRHVTNVENLIDDAIDLLKRHINRDSPDRAALVASVNRLVPVATKLVVGHFERTLWARAMARVADDPAIAAGAILVRTRRLDRRVDPLAVYASASADRLRTVWLRPDHGVGLAAIGEVEAIEPTGNDRFSAASAARAVLAARIRRDMPSEGPAPVLVGGFSFAANREAAGNHSQPPGDNGSGSPHRGPSWRGFGDCRMVLPEL

>Sequence_13: AAAAEAGMTVGPVAIRDAYPHSRLQDEELVELTVGANGALALLSVSSTGRTADSLAQCLDERIGRGRSGAGWRLETLVHRIQTRAARWHEDADARGEPWLHTPGGESFAAGDDCELCDSADRAPYVR

>Sequence_14: AAAAEDGLRYEDPERIEILRDVLKAALDDKLITFRELNLLRTLRDSLGLSEAVFRIVLAQLNHFPQRGNVLHTPSECRDVLNKLQWGGIVFHCNKADDAPYVIPDEIRASV

>Sequence_15: AAAAEDSGVAVQYDTAPERFRDLLRALHARTGQRVAVLVDEYDKPILDALETPDVARANRDYLRGLYSTIKSSDAHVRFTFLTGVSKFSKVSLFSGLNNLTDITIDPRYSAVCGYTEADLDTVFARELAGLDRERIRDWYNGYNWRGAERVYNPFDILLLLDKREFGC

>Sequence_16: AAAAERTVDAGGRLCVTGFLEPHIHLDKAQINDDVRPNASGTLDEAIEIIWARKRAYTVEEIAARAVRTIRSAVAHGVTRFRSHVDVDSIGGLRPLEGVLEARERTRDIADVQIVAFPQEGIFKDKGT

>Sequence_17: AAAAFADCRGVRDADDIAAGLCDMSDVVARLKYPGTSSVVQEVFDQLFYGPDLYKKIAGRESRFTEPGLIERENILLDGATREWLTRKFGPRIGMVTGRGRESARHTLGD

>Sequence_18: AAAAFAPTGVFAQSVAFTPPKESTPVKVAAGIAHAMKQEGIEPPEVPRPRVFVAYLGAEAKDAAVLLAQELRTAGIRTAALWEDRSLKAQMKQADRLGAGYVLIIGEDEVKDEVVTVRR

>Sequence_19: AAAAGAAGVAGTRSVRLSSKALPMALPRQMRSVRPERNDQEIVLYVRSIPFLTPREITATIREAGYVGQDRAVTAVSLMAYRHVSRLRKMHLEGISRDLLPDKTNLLLVGPTGCGKTHLVELLFQHILHLPTITV

>Sequence_20: AAAAGVAVVSEKPMATSVEQAEQVLAAVRRHGVPYAVVHNFLFTPGMERARALLAEGAVGTPFFGRAKSLFNKTDDQADPDTVWRASLAAGGGAINDTAYHEIYLLETLVGSPVRYVEARVATQHFSFEVDDLALLLLEHDGGEVAPVAALRPRRLRRQLRLPPLARTARIVPQPAPAAHHPHAAGRALHLGQVGALAGVPARHAPGGGHGG

>Sequence_21: AAAAGWPGVKRVLTAADVPSNRVGIYPDLRDQPVLAEGVVRYRGDPVVALVGTRDAVTAVRDDDLPIRYAPLPAVLGAGKAGAPDAPLVHDDMADNRLIEGGVERGDAKAALDDCAATAEGRFETAFVEHAYIEPEAGWARRVGDR

>Sequence_22: AAAANADPVEFRLRLLRGTDEDDATFRRARSIAVVEAAAEAYGWDARPSPRAPRDGRILTGRGISHAYRIQTTVATIAEVEVDRESGRVWVKRMVCAHDCGL

>Sequence_23: AAAAPAGLYAQTSMRIEKGFAAMGHELDGDIGPVEAGLDRLCARNKRFVGSDALAGRRKTGGRTLATIVLDDPDALPLGHEPVRFGGRIAGQTTSAAFGYRIGRPLALAHVAGAPDGAEVEVDIAGDRHPGRLRFAPAN

>Sequence_24: AAAARADPVEFRLRLLRGSTDDDDAYRRARSIAVVEAAAEAYGWDPRPSPKPVGSERILTGRGIGYAYRARTTVATIAEVEVDRESGRVWVRRMVCGHDCGLVINPDGLESTVQGNLLHGISRTLYEEVRFDGEKVTSVDWR

>Sequence_25: AAAARDDPYAMVDALPEHWDRFRRDSVTDLVRRVYRDVKARQPHLVVSAAVLPDWRAAAKWQFQAWSSWLAEGILDVAVPLAYTEDDQQFRAWIDAAASAAGAPGRVWAGVGAYRNPVERTVRHIDLARAA

Continúa en la siguiente página

Tabla 34 – Continuación de la página anterior

Secuencias

>Sequence_26: AAAARDGDGIDSDPYDVGDGTPASAPSSFHGTHVAGTVAAATNNGIGVAGVTWRGKVMPVRVLGAGGGRSYDLAQGIRYAAGLANDSGT
LPAARADVMNLSLGIPNEECLPLGVDRYTRDALADAMANGVVVVVAAGNDACHEPDPMSRVDGVISVSATDYTGLAPYSNYGRTVDVAAPGGHRGIDRDGDGYN
DGVLSTLA

>Sequence_27: AAAARMHGVEQGVDLRGVTILAFGGAGPVHACGVADLLESDKIVFPPNASVLSAFGTLVSPVRIDLARSMPRLLRSVEEAERDQLLDELRGEG
RRVLLAAGVPQSEVRFRYGLDARYAGQGNEITVWLGDG

>Sequence_28: AAAASELRQNFAEDEGFLVPAVDWQRTSRRVLTLERVEGIPIDERDALIAAGHDPHEVLGKAASAFFNQVFRDGFFHADLHPGNLLVDADGNVV
AVDFGIMGRLDTQTRRYLA

>Sequence_29: AAAASQTERIRLGTGVALLALSDPIRAAEDAATVDIISGGRLEFGIGRGSIAEHFAGFNVRMADRAGRFNEALAVIQQAWADGPVNFEGDHFSY
RNVQVVPKPIQRPGPPLRMAANS

>Sequence_30: AAAATACVTSSSLVTSASVKMPPRSAATSAPFERPRSRMAIFAPRACSSRAVASPRPEPPPVTIALSPLMSMSALRSRCGAGCVLPRTPSRGD
SGGGMRRTLREGRRPGSTVWGVMMATDADEAAASWSEAVDEIERRRELAGRMGGEERVARQRAQGKLTIRERIDQFLDAGSFFEIGGMAGTGDYDGDGRLTDFI
PAGYVLGLGTVDGREVCVGGEDFTVRGGSGGGGKGPGRWLQPAALQRRIPLVQFADGAGASARSYDDAGRMHLPDGNQWARDAELLSKVPVCSAIVGPSAGHVA
GRAVLSHFSVMVRGQGQVFAAGPPVVKRAIGEDLTKEELGGVGVHVRTSGVVDNEA

>Sequence_31: AAAAVERGLSYLAVTDHGEDLGLNGVSRERLLAQRVELRALDGVHGGLRLLHGVELNIGPDGGLDYDAAFRAAFDWCVAAVHSAFELDRA
AQTRRIVRAMEDPSVNAIGHLSGRMIGRRPGIELDLDAVLAAAERTGTAIEINGGLNRLDASDDVLRRARARNVVILVNSDAHHVSDFERLQWGVQQAQRGWVDRD
RIANTWPAERFL

>Sequence_32: AAAAVPADVLPGQLLQPGIELHAVGVQLRNCVPTNEVCTQTGRMPCRSGSKFSLFDQDDVLPTFLRKMVGQARSHRSATDYGNARIRGHCR
NSPHSGPYCRVL

>Sequence_33: AAAAYHLEQILGSPWLEVDLIKSTRLKVVIDAGNGAGSMISPSLLRELGCEVIELNCVADGHFRRSAEPTPEALDELCKIVLASEADIGFAHDGD
ADRLVLVTERGIPLNGEWTLAFVVDLILGKTKGDVVATVSTSRMLDDITVNHGVTLHRTKVGVGWVVEKMHEVNAVIGGEGTGGVIYPNVHHTTDGITSIAAIVQYLA
ESGGTM

>Sequence_34: AAACGVLLAAWPAVAAAQPAADGAQSPRNASYAIDVELDAARRTLTGREVLTWRNIAPVATAELQFHLYYNAWKNTQSTWMRELRRRGRRGQ
GLHTRPEADWGWIDVTAIRLLGSGGEPAADLTDRAAYIAPDDGNS

>Sequence_35: AAACHSPALAPPRDLDHAAAQIRARAETAAEFYAAALSRRTRARLAADFASVGELLRDRVVVALDRRERRRLRRFLARLAGTPSQAAQCSAAA
WLRLADASWLARLRELIAAHPNAAADVVLRVDTPLGDISFGGRGDAR

>Sequence_36: AAADDDLGAIHHPGVDPQGQIAGDPERCHAADLHSGQLLCLVGRGERGPSHAQPLAHHPVHVQARRRQHQAHHVPRRIGLPDHHDGIGGVL
RGVAVRLAEVGGVGQIGIAGHHLVGQPQRIELCENRIGHAPTLEESRAVRGERRRHRWTVDSQPPECAPVGPARRL

>Sequence_37: AAADESTTALAGEGVGNVGRDEEPADPPTLDTLAREGARRMLMAALEAEVTEHIERFAEVRDADGRRLVVRNGQAQERTVICGAGTWQVRA
PRVNDKRVDAEGNRERFTSRILPPYMRRSPQVSEVLPLLYLRGLSTGDFREALPALLGEDAAGLSPTTISRLTAAW

>Sequence_38: AAADGGGGPSGPQERARVLFLLRRRGGLRQSDMDEARVREAVELLLAEVGEDPTRQGLVRTPDRVARALQFLTSGYDTDIADVVNDALFAIDGS
EMVIVRDIDFYSLCEHHLLPFFGRCHVAYLPRDQVIGLSKIPRLIEMFSRRLQVQERLTNQIASALAEVLDPLG

>Sequence_39: AAADILEQLRAEGNSAAPENTLWRRLGDPFMLRREYGRRTAEDLAELFGGAFVDGLATLPIGEWAGPVESAYGYHIVRLSVRTEARPAEFEDV
RRRVLDDAQRDARDAANRAAHERLRERYDVEIEEI

>Sequence_40: AAADKPPVIAGFLPDLAGAVVVTTPSEVASNVVQKSIGYATELGITVLGVVENMSQYRCPSCGAENELFEGNTQGMCEALNLPLLGRIPFDRNF
AKTFDKGTPLLDGEYPTNRRYTDIVGRIQSMLDYKRVLAEKL

>Sequence_41: AAADLQETASNAGYENLLRLLHLLLSSETVVRRADAPGLALEVALLRAAELPRLVRIERLAAGLPAEAEPPAGRPGAPTPRGDTAEPPAAISSP
DASASVVTRLESKSPALAGLLDFHHAHLEARGNALVIEAGPELQRELRDEARTAQLDDAVRFVLGPDAAWKVTPAPAGRTGDASPATAADRPAAVDRPAADAAPRPE
RRRPAPERKRARRPQGPDSEARRIAEADPGVRAVLDIFGGRVLDVRPKST

>Sequence_42: AAADLVSNGIVTNEQMRAAHPERIAAVVRAFDKGLRATIDNPPAAYLHSAAYIESLPLSEALAEALRKLAEEQAEFLARKPGRGEIAATRARQYQAL
ANDPALAHELTQFRVLLATIALWE

>Sequence_43: AAADPRIDHLRARTRVHEDPAYTEAYLDPARRAIGNRIQVRFRDGTRTRPVAVDYPLGHPRRRAEAEPHLAAKFARNTATRLPERSAAALRRLW
DDSE

>Sequence_44: AAADQAAILLDRSHEGRILVSGDDRLNLVNRMSTNDLGALAVNDGRATIFTDANARILFRAECFERPEGLLLISQPGQGQALASFLQRNIFFGD
KATVNEQSAATAQFGLHGARADEIALALDSHLSDLPAMGSCDVTLAECPAILTRRKPLSGGHWSLICAAEAATVVHEQLLELGRLLGLQPAGSLAYNILRIRSGYPGGA
ELSTDYIPLEVGLWDEISFSKGCYTGQEIIARMESRERVAKAMVKVDLTTFAPARILGPQGLQRCW

>Sequence_45: AAADSAAASAHKLGGPVGVGALRLRTGATLEFLPRGGGQQGGRRPGTENVAGMRAFARQARTLAQDAAGHLRRARQLCERLIAGARRLRL
EVVPRQRLQHPEHYSPYITCLAAPGVPSEVLVRILSDQQIYVSRGSACASGGPKTSPILQAMGVPAAVAGGAFRVSTGWTTTEADIDRLLD GLAAALPALRAIAQ

>Sequence_46: AAAEAAGRGSVRRFRIGGKTDRLHGEALEVECRVLDLRDGRFHEPEVRHGGVRHYDQGRTAVVESATGLVIMLTSKRMVPFSLHQLTDFGIDP
KTFRFIVAKGVNAPLAAYREVCPSIVRVDTPGSTAANMTSFAYRHRRRPMYPFEP

>Sequence_47: AAAEALDVDRLRESFPILGREVHGHPLVYLDNAASSQRPDVVIGAVSDCYRTYYANAHRGVHALSEQSTDAYEAARVKVAGFIGAASSAELVF
TRGTTESLNLVAQSYARPRLEPG

>Sequence_48: AAAEANLQSGSVDILLISPERLANQRFRSELLPEILQGSGLLVIDEAHCISDWGHDFRPDYRRIARVLDFLPGVPVLCCTATANNRVVEDVKAQL
GSRLTTVRGPLSRTGLRLQVFDIPSQAERLSWLAEVVPRLDGTGVIYCLTVADTRRVAEWLVSTGIAAVAYSGATDP

>Sequence_49: AAAEDMIRREGARPAFKGYRGFPASVCASVNHQIVHGIPNDQPLKDGDILSLDVGVYYKGFYGDSAWTFCVGEISPELNKLLDVTRESLYRGI
EKVQVGNRISDISHGIQEYVEAFGFSVVREFVGHGIGRSLHEEPQVPNYGKPGKGPRLKPGMVLAIE

>Sequence_50: AAAEDRFGRLDVLVNNAGISIHAITLDASLEVWRKVFDVNATGTFLCSQRAGRIMVPRGGGCIINIASVYGHVGVDPSVSHPPGFRSREDLAYS
ASKGAVISMTRALATYWADYRIRVNAISPGMIRTKRLETAVPGDTWSRLIQRTPLKRPGTGEDLKGAVIYLASDAASFVTGQVLVVDGGW

>Sequence_51: AAAEEAREFVDELRTGRSLVEVAGEFALEVRDAGPFTRNQFVFGLGRANAAVGAAFGLEVGEISDVVVANQNVFVIERTASQPADSLAWVA
QIDLQRQQELLITRQTRLNLWIEALRKSADIVDRREQVLRPAD

>Sequence_52: AAAEECYRHAQASAKEEVARAHRLEQEGDHRTAAGLFASHGRSERAAHNLERAGQLQEASRQWRRASNQERALSCEARHYEAAGNFKAAA
KRWQQLGDEEATLRTWERGRQYRQLAAFYREREVSGEAARYLKLAGDHAAAAVEFRRAGLLELAAQEFEQARDYKRAAALYRRLGDSAALLRCLGSMHIDGAHEA
ALVYEEQRNWTKAVEHFRRYAE

>Sequence_53: AAAEGGHIGSMNQLIELGSGINTRASRTGWTALLIAANQGKTEAAKLLMEKGANVNSKTKDGLPCLMLAINEGHADTVEALIQGGARVEFRLNAD
WTPLMHAASEGKTEIVRSLLAAGADVKATNHNGWTPLMLAVAEKKPDVIRALVEHGAEVNVKGNDGRSALK

Continúa en la siguiente página

Tabla 34 – Continuación de la página anterior

Secuencias

>Sequence_54: AAAEHTHLLVEAKKLAYADLHAHVGDPEGEPLRNSGGVPLAELLDKAYARERSRLIDPGRAADGAAPGGIPAGGGAVGSDTVYLAVVDGDG
NAASFINSLFAPFGAAIAGGDTGIMLHCRGAGFTLEPGHPNEYRPGKRPFHTIIPGIVLERGRLELCYGVMGGPFQPQGHVQLLTNHYDHGLPLQASIDRPRWRHTD
GLDLLCERGMEPWTVEGLAALGHRVRRAGGAEFGGAQAIGVDRHGTLHGASDPRKDGAALGY

>Sequence_55: AAAEIARVDPERVLDLGGGTGALAEAILLAAPRAVVELIDVDGEMLAVARERLVPFGSRARFREASFDDPLPEADAVAASLALHHVPTMDEKRR
LFVRIRDCLRAGGVFVNADVTMPAGPDAREAVWRGWAAHLVSCGIPEERAYRHFAEWQEE

>Sequence_56: AAAEPMARAFGAPLVVNSASRDGSLREAADKRDVPVIVYEAGEALRFDETAIRTGLRGVIGVMEHLGMVPTQKRGSETKAPVILRESSWVRAPA
SGVVRVRQPIGAEVEGGELLAVVSDPLGES

>Sequence_57: AAAEQVAAAMTAGGAAPPYADCNAIAPATTQRIAAVIEAAGAVYIDAGIIGGPPAGGNGPRFYASGPAARLLEPLDGGGLVVRDLGGAVGQAS
ALKMCYAAVTKGTSALQFAQLAAASRLGVDEALAAELAESQGATYASMQRSLPGLPAKAPRWIEEMRQIAATFEAVGVPGGFHHGAAALYELLSAT

>Sequence_58: AAAERGAYDAKVAAREKRRGRAKGKHPKPPDETPAAAEQSNLSDPDSRLMRKSKRHEYRQAYNAQAAVDVGGSQLIVGARVGQCASDRNELV
ADVAAIPAVLGRPETVLADNGYANGAEVASLEASGIEVLV

>Sequence_59: AAAERLDAPGRLREALDREHVGRGREKDQVLDYLVARQAAAERGAAGAHAGAAKTLCLCGPAGTGRTAFARALAAALGRRFVRASLAGVEKP
AGILGAARSAPDAGPGRLLDALRRLGIDASVRTVDPTQFRSRLRGFAYDAAINDYWATPTLGTYLKSFFHSSGADA

>Sequence_60: AAAERWRAEIQAHATVDPFIVDLIRSFPRDAHPMGILISALGAMGTLYPEAKEVLDSDVRRRQVYRLLGQLPVVAAHIYCHGQGVEAEEPQA
DDGYIGNFLRMVRNFGL

>Sequence_61: AAAEVDGVRFDGVDLFLSAPHVDIDSTGDDLKRLAEKVQSLGLVIGSVVAPVWPPTGGGSAMGDESERGRFLSMVEKACVIARRLRELGVR
PYGVVRIDSSVDPGTWAADPEANTRRIAETFREACARAGDQGELLAAEGEVCWGGMHSWKEMLKLLELV

>Sequence_62: AAAFDGGVNFIDTANVYEEGRSETFVGKFLQGRRYDVILATKGNGRLGQGPNDRQVSRKYALQAIDDSLRRLNTDYIDLYQIHDFPLDVPIDEFV
LAYSDIVEAGKASYIGVCNFAAFQLVEALWEAEKRGARLFNCLQTRYSLLARDPERELFPVCEEYG

>Sequence_63: AAAFDLLRQDAVDEVVIADLRPDALHPALRPHLGGKLATLCADAASGAASSAASGAAPDDASHDDIRRAMEGVAGVLCALPYHFNYRMARM
AVDAGVHFTDLGGNTAIVTRQRGLDAAARKKGVSVAPDIGLAPGMVNILAQAGIEHFDTVHSVRLWVGGLPRSPRPPLNYQIVYALE

>Sequence_64: AAAFEETLRTYADKARAKYAVRVDAQDEDSLKSLIGDLLRDVGSGWDMETEWRSEARAEGDIGRPDLGITANRLLCGHIELKAPGRGARPEQ
FRGRDRDQWNRFQALPNLIYTDGSEWSLYRSGERTARIRISADVTHDGAAGLDAAALGKIEGLLRDFLMHEPVTPSTAQGLAEFLAPLARLLRDEVREALDVEGSV
VGQIADDWRGVLFGDAET

>Sequence_65: AAAFLPDGTEISIQLAALLRDLAGTDRRYRESVGDTAATWEAVEGGGVVINETMANRYNVGVGRALRLLTDGGDREFTIVGVTYDFDVHPTV
LMADTVYRRHWNDTGISTIALIVQPGVNVDETVNALRRTFAGRAELVIRSNRGVRDNALEVFDRTFAITVALRLLATFVAFVGILSTLM SLQLERAR

>Sequence_66: AAAFVQLQQDFGMTQQAVASRVGRSRAAVTNATRLLQLTAEVQELVASRQLSAGHGRALLAIADASEQGRLARQVVSEGWSVRELETHVRPP
ARSGDEPVASRPTDVAPPAKQEPVAANASNSARLELESLLAEFLATTVRVEIGDRKGRLTVEFADL

Anexo C. Resultados en las métricas de desempeño de los modelos por ensamble en conjuntos de prueba y externos.

Tabla 35. Resultados obtenidos al evaluar los modelos por ensamble, construidos para cada actividad, usando los conjuntos de prueba y externos propuestos.

Actividad	Ensamble	SN	SP	ACC	MCC	SN EXTERNO
Antibacteriana	WR_NO_25 WR_ES_29 WR_AC2_30 WR_ALL_24	0.883	0.996	0.925	0.853	0.880
Antifúngica	WR_NO_21 WR_AC2_23 RA_AC5_10 RA_ALL_10	0.922	0.940	0.933	0.857	0.650
Antiparasitaria	WR_NO_28 WR_ES_29 WR_KH_23 RA_ALL_40	0.846	0.645	0.737	0.496	0.810
Antiviral	WR_NO_21 WR_AC2_23 RA_AC5_10 RA_ALL_10	0.922	0.940	0.933	0.857	0.650

Anexo D. Error cuadrático medio.

Para analizar el desempeño de los clasificadores en un proceso de validación cruzada, es necesario observar una medida de la variabilidad en el desempeño del modelo en cada iteración de esta. La métrica más utilizada es conocida como la raíz del error cuadrático medio (RMSE, por sus siglas en inglés), la cual expresa la desviación estándar de las predicciones obtenidas, cuanto más pequeño su valor, menor varianza en el desempeño del modelo. (Chai y Draxler, 2014). Esta métrica se calcula como sigue:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(f_i - y_i\right)^2}, \tag{27}$$

donde f_i es el valor observado en la predicción, y_i es el valor esperado y n el número total de casos.

La métrica RMSE es utilizada, generalmente, cuando las predicciones son valores continuos, sin embargo, puede aplicarse a los resultados de un proceso de clasificación binaria, donde los valores predichos son discretos. Lo anterior se debe a que es posible obtener una distribución de probabilidad de las predicciones, y se pueden utilizar las probabilidades de cada predicción para calcular dicha métrica con la siguiente fórmula, conocida como la métrica de Brier (Brier, 1950) equivalente a RMSE:

$$RMSE = BRIER = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(P_{f_i} - P_{y_i}\right)^2}, \tag{28}$$

donde P_{f_i}, en una clasificación binaria, es la probabilidad en la predicción de pertenecer a la clase positiva, P_{y_i} es la probabilidad real (1 o 0) y n el número total de casos.

A continuación se muestran los valores de la métrica RMSE de cada clasificador clásico construido, obtenidos bajo un proceso de validación cruzada de 10 pliegues, cabe señalar que la métrica esta implementada en el software Weka bajo la definición de Brier (1950).

Tabla 36. Raíz del error cuadrático medio obtenido por los clasificadores clásicos bajo un proceso de validación cruzada de 10 pliegues.

Conjunto de entrenamiento	Modelo	RMSE
AntiAntimicrobiano	PseACC	0.304
	WR_AMP_131	0.302
Antibacteriano	PseAAC	0.258
	RA_NO_10	0.299
	RA_NO_20	0.296
	RA_NO_30	0.285
	RA_NO_40	0.280
	WR_NO_25*	0.279
	RA_ES_10	0.328
	RA_ES_20	0.314
	RA_ES_30	0.299
	RA_ES_40	0.287
	WR_ES_29*	0.287
	RA_KH_10	0.347
	RA_KH_20	0.301
	RA_KH_30	0.301
	RA_KH_40	0.303
	WR_KH_27	0.302
	RA_AC2_10	0.337
	RA_AC2_20	0.307
	RA_AC2_30	0.293
	RA_AC2_40	0.291
	WR_AC2_30*	0.291
	RA_AC5_10	0.351
	RA_AC5_20	0.317
	RA_AC5_30	0.317
	RA_AC5_40	0.314
	WR_AC5_23	0.311
	RA_ALL_10	0.321
	RA_ALL_20	0.301
	RA_ALL_30	0.292
	RA_ALL_40	0.289
	WR_ALL_24*	0.288
	WR_EM_61	0.261
Antifúngico	PseAAC	0.229
	RA_NO_10	0.243
	RA_NO_20	0.227
	RA_NO_30	0.224
	RA_NO_40	0.221
	WR_NO_21*	0.221
	RA_ES_10	0.242
	RA_ES_20	0.232
	RA_ES_30	0.230
	RA_ES_40	0.228
	WR_ES_25	0.225
	RA_KH_10	0.250
	RA_KH_20	0.239
	RA_KH_30	0.242
	RA_KH_40	0.240
	WR_KH_25	0.242
	RA_AC2_10	0.271
	RA_AC2_20	0.249

Continúa en la siguiente página

Tabla 36 – Continuación de la página anterior

Conjunto de entrenamiento	Modelo	RMSE
	RA_AC2_30	0.249
	RA_AC2_40	0.249
	WR_AC2_23*	0.251
	RA_AC5_10*	0.265
	RA_AC5_20	0.255
	RA_AC5_30	0.252
	RA_AC5_40	0.247
	WR_AC5_26	0.248
	RA_ALL_10*	0.243
	RA_ALL_20	0.241
	RA_ALL_30	0.232
	RA_ALL_40	0.231
	WR_ALL_25	0.230
	WR_EM_21	0.226
Antiparasitario	PseAAC	0.321
	RA_NO_10	0.384
	RA_NO_20	0.386
	RA_NO_30*	0.367
	RA_NO_40	0.366
	WR_NO_17*	0.354
	RA_ES_10	0.407
	RA_ES_20	0.399
	RA_ES_30	0.385
	RA_ES_40	0.308
	WR_ES_13	0.366
	RA_KH_10	0.432
	RA_KH_20	0.402
	RA_KH_30	0.385
	RA_KH_40	0.374
	WR_KH_20	0.358
	RA_AC2_10	0.411
	RA_AC2_20	0.390
	RA_AC2_30	0.371
	RA_AC2_40	0.370
	WR_AC2_17	0.370
	RA_AC5_10	0.427
	RA_AC5_20	0.403
	RA_AC5_30	0.393
	RA_AC5_40	0.392
	WR_AC5_20*	0.380
	RA_ALL_10	0.401
	RA_ALL_20	0.382
	RA_ALL_30	0.382
	RA_ALL_40	0.378
	WR_ALL_27*	0.375
	WR_EM_43	0.354
	PseAAC	0.355
	RA_NO_10	0.402
	RA_NO_20	0.377
	RA_NO_30	0.371
	RA_NO_40	0.366
	WR_NO_28*	0.366
	RA_ES_10	0.407
	RA_ES_20	0.394

Continúa en la siguiente página

Tabla 36 – Continuación de la página anterior

Conjunto de entrenamiento	Modelo	RMSE
	RA_ES_30	0.389
	RA_ES_40	0.381
	WR_ES_29*	0.382
	RA_KH_10	0.429
	RA_KH_20	0.412
	RA_KH_30	0.406
	RA_KH_40	0.400
	WR_KH_23*	0.400
	RA_AC2_10	0.414
	RA_AC2_20	0.403
	RA_AC2_30	0.396
	RA_AC2_40	0.390
	WR_AC2_29	0.392
	RA_AC5_10	0.441
	RA_AC5_20	0.422
	RA_AC5_30	0.405
	RA_AC5_40	0.398
	WR_AC5_30	0.398
	RA_ALL_10	0.407
	RA_ALL_20	0.392
	RA_ALL_30	0.390
	RA_ALL_40*	0.387
	WR_ALL_25	0.389
	WR_EM_69	0.369

* Modelos seleccionados para el ensamble.