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

Doctorado en Ciencias en Ciencias de la Computación

Métodos de selección óptima de descriptores moleculares para la clasificación de la actividad antimicrobiana y el diseño de nuevos péptidos

Tesis

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

Presenta:

Jesús Armando Beltrán Verdugo

Ensenada, Baja California, México

2019

Tesis defendida por

Jesús Armando Beltrán Verdugo

y aprobada por el siguiente Comité

Dr. Carlos Alberto Brizuela Rodríguez

Director de tesis

Dr.Israel Marck Martínez Pérez

Dr. Hugo Homero Hidalgo Silva

Dr. Gabriel Del Río Guerra

Dr. Ubaldo Ruiz López

Coordinador del Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez

Directora de Estudios de Posgrado

Resumen de la tesis que presenta Jesús Armando Beltrán Verdugo como requisito parcial para la obtención del grado de Doctor en Ciencias en Ciencias de la Computación.

Métodos de selección óptima de descriptores moleculares para la clasificación de la actividad antimicrobiana y el diseño de nuevos péptidos

Resumen aprobado por:

Dr. Carlos Alberto Brizuela Rodríguez
Director de tesis

La resistencia a los antimicrobianos es un problema de salud mundial que pone en peligro el éxito del tratamiento de infecciones comunes. Los péptidos antimicrobianos (AMPs) son una alternativa prometedora para sustituir a los antibióticos convencionales en la lucha contra patógenos multirresistentes. Estos péptidos son parte del sistema de defensa innato de la mayoría de los organismos vivos, para muchos de los cuales se dispone de datos de sus transcriptomas. Sin embargo, considerando la vasta cantidad de datos transcriptómicos, el espacio de los péptidos sintetizables y la limitación práctica de la evaluación *in vitro*, descubrir nuevos AMPs se torna un desafío complicado. Motivado por este desafío la tesis aquí propuesta plantea utilizar estrategias computacionales para ayudar a descubrir y diseñar AMPs. Para poder identificar computacionalmente a los AMPs se requiere un conjunto apropiado de descriptores que ayuden a discriminar entre AMPs y no AMPs. Desafortunadamente, dado que existen miles de estos descriptores, una búsqueda exhaustiva de todas las combinaciones posibles es inviable. Por lo tanto, en este trabajo se proponen dos enfoques para seleccionar automáticamente los descriptores moleculares que representen a los péptidos para poder clasificarlos en AMPs y no AMPs. (i) Utilizar un filtrado basado en la ponderación de los descriptores, donde se asignen pesos a estos, de tal forma que secuencias AMPs tiendan a estar separadas de las que no son AMPs, mientras que las secuencias de AMPs tiendan a estar cercanas entre sí. (ii) Utilizar un enfoque de envoltura basado en un algoritmo genético donde cada péptido es representado por los descriptores seleccionados (*i.e.,* con una longitud variable) y una función de aptitud que considera el coeficiente de correlación de Matthew de un clasificador inducido y el número de características seleccionadas. Los experimentos computacionales muestran que estos enfoques reducen sustancialmente el número de descriptores requeridos mejorando el rendimiento de clasificación con respecto al uso de todos los descriptores moleculares inicialmente disponibles. Además, el desempeño de los modelos generados es competitivo con las herramientas de predicción de AMPs disponibles actualmente. En particular, el enfoque basado en ponderación de características produce modelos de clasificación que superan a las herramientas disponibles para la clasificación específica de la actividad antibacteriana. En lo que respecta al diseño *in silico* de AMPs, los trabajos existentes se enfocan en la exploración del espacio de secuencias de péptidos principalmente cercanas a los AMPs conocidos, por lo tanto, solo se explora un área limitada sobre todo el espacio molecular, lo que causa una pobre diversidad en los péptidos diseñados. Por ello, en este trabajo exploramos un enfoque para el diseño de nuevos péptidos utilizando dos modelos basados en aprendizaje profundo: el primero

es un modelo generativo que utiliza una red neuronal bidireccional recurrente para la generación de nuevas secuencias; el segundo modelo es de clasificación y utiliza una red neuronal profunda para determinar si los péptidos diseñados son AMPs o no. Estos modelos proponen generar secuencias nuevas que no sean similares a nivel de secuencia a un conjunto de AMPs de entrada y a su vez que sean similares al conjunto de entrada en el espacio químico. Entrenamos ambos modelos con la colección de secuencias más grande hasta ahora utilizada para la clasificación de AMPs (*i.e.,* 43403 secuencias). Los resultados para el modelo de clasificación resultante muestran que permite clasificar con una exactitud de 91.80 % el conjunto de prueba y tiene una área bajo la curva ROC de 0.97, superando así a los métodos de aprendizaje de máquina de última generación para la clasificación binaria de AMPs. Finalmente, los resultados preliminares para la generación de secuencias muestran que es posible producir un modelo generativo utilizando una red neuronal bidireccional recurrente de memoria a largo y corto plazo con un error mínimo en la validación, mostrando así la capacidad para poder construir nuevas secuencias de aminoácidos con alta probabilidad de ser AMPs.

Palabras clave: Péptidos antimicrobianos, ponderación de características, descriptores moleculares, representación peptídica

Abstract of the thesis presented by Jesús Armando Beltrán Verdugo as a partial requirement to obtain the Doctor of Science degree in Computer Science.

Methods for Optimal Selection of Molecular Descriptors for Antimicrobial Activity Classification and Design of New Peptides

Abstract approved by:

Dr. Carlos Alberto Brizuela Rodríguez
Thesis Director

Antimicrobial resistance is a global health issue that compromises the effectiveness of current medical approaches on most common infections. Antimicrobial peptides (AMPs) are a promising alternative to conventional antibiotics in the fight against multi-resistant pathogens. These peptides are part of the innate defence system of most living organisms, for many of which transcriptome data are available. However, considering the vast amount of transcriptomic data, the space of synthesizable peptides and the practical limitation of *in vitro* evaluation, discovering new AMPs becomes a difficult challenge. Inspired by this challenge, this thesis proposes to use computational strategies to help in the discovering and designing of AMPs. To achieve this, an appropriate set of descriptors is required to help discriminate between AMPs and non-AMPs. Unfortunately, given that there are thousands of descriptors, an exhaustive search of all possible combinations is unfeasible. Therefore, in this work, we proposed two approaches to automatically select a peptide representation, based on molecular descriptors, that efficiently performs the classification between AMPs and nonAMPs. (i) Using a feature weighting approach, where each descriptor has an assigned weight, in such a manner that AMPs tend to be far away from non-AMPs, whereas AMPs tend to be close together. (ii) Using a wrapper approach based on a genetic algorithm, where each peptide is represented by the selected descriptions (*i.e.*, a variable length representation) and a fitness function that considers the Matthew correlation coefficient of an induced classifier and the number of selected descriptors. Computational experiments show that these approaches substantially reduce the number of descriptors, thus, improving the classification performance of the case where all initially available descriptors are used. Also, the performance of the generated models is competitive with the tools currently available for AMP clasification. In particular, the feature weighting approach produces classification models that outperform the tools available to classify antibacterial activity. Regarding the in silico design of AMPs, existing work mainly focuses on the exploration of peptide sequences close to known AMPs. Therefore, only a limited area over the entire molecular space is explored, causing poor diversity in the designed peptides. For that reason, we explore an approach to design new peptides using two deep learning models: the first, is a generative model that uses a recurrent bidirectional neural network for the generation of new sequences; the second, is a classification model that uses a deep neural network to determine whether the designed peptides are or are not AMPs. These models attempt to generate new sequences that are different, at a sequence level, from a set of AMPs given as input, and, at the same time, that are similar to this set of AMPs in the chemical space.

We trained both models with the largest sequence collection ever used for the classification of AMPs (*i.e.,* 43403 sequences). The results of the classification model show 91.80 % of accuracy in the testing set and it has an area under the ROC curve of 0.97, thus outperforming the latest generation machine learning methods for binary classification of AMPs. Finally, preliminary results for the AMP sequence generation show that it is possible to produce a generative model using a bidirectional recurrent long and short term memory neural network with a minimum error in validation, showing the ability to build new amino acid sequences with a high probability of being AMPs.

Keywords: Antimicrobial peptides, feature weighting, molecular descriptors, peptide representation

Dedicatoria

A mi esposa, mis padres y hermanos

Agradecimientos

Agradezco especialmente a mi esposa Franceli Cibrian por su apoyo, inspiración y comprensión durante el proyecto, gracias por alentarme a terminar este trabajo de tesis.

Agradezco a mí director de tesis Dr. Carlos Alberto Brizuela Rodríguez por su guía y enseñanzas. También, a los miembros del comité de tesis, el Dr. Israel Marck Martínez Pérez, el Dr. Hugo Homero Hidalgo Silva y el Dr. Gabriel Del Río Guerra por sus comentarios y sugerencias durante el desarrollo de este trabajo.

A mis amigos y compañeros de Ciencias de la Computación con quienes conviví y de quienes recibí su apoyo y consejos en momentos importantes, además de hacer mi estancia en el posgrado un momento agradable.

Al Centro de Investigación Científica y de Educación Superior de Ensenada por permitirme estudiar este posgrado, y al personal administrativo de esta institución, en especial a Angélica Lomelí y Karina Ortiz por su ayuda y disposición durante cualquier trámite administrativo.

Finalmente, al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de doctorado.

Tabla de contenido

Página

Tabla de contenido (continuación)

Tabla de contenido (continuación)

Tabla de contenido (continuación)

Lista de figuras

Figura Página

Lista de figuras (continuación)

Lista de figuras (continuación)

Figura Página

Lista de figuras (continuación)

Figura Página

Lista de figuras (continuación)

Figura Página

Lista de tablas

Tabla Página

Lista de tablas (continuación)

Capítulo 1. Introducción

En los últimos años hemos sido testigos del gran avance que han tenido las técnicas de aprendizaje de máquina con un espectro amplio de aplicaciones en problemas del mundo real. El impacto se ha dado en el área de análisis y reconocimiento de patrones en textos (Petasis, 2012), imágenes médicas (Litjens *et al.*, 2017) y visión por computadora (Sebe *et al.*, 2005), por citar algunas. El área de bioinformática también ha sido beneficiada por el desarrollo de estas técnicas (Baldi y Brunak, 2000). Por ejemplo, el desarrollo de las ciencias ómicas tiene como sus pilares tanto a algoritmos para la representación compacta de datos para su eficiente recuperación, así como a algoritmos de aprendizaje de máquina para la identificación de regularidades dentro del universo vasto de datos que estas tecnologías generan (Lin y Lane, 2017).

A continuación, presentamos las motivaciones tanto biológicas como computacionales que dieron origen a este proyecto de investigación.

1.1. Motivación biológica: el problema de la resistencia a los antimicrobianos

Actualmente, la resistencia a los antimicrobianos representa un problema de salud mundial, alrededor de 700,000 personas mueren al año ocasionados por cepas resistentes a los medicamentos relacionadas con infecciones bacterianas como en tuberculosis, virales como en VIH, y parasitaria como en malaria. Desafortunadamente, de no tomarse medidas para mitigar el problema de la resistencia a los antimicrobianos, se estima que para el año 2050 el número de víctimas mortales ascendería a 10 millones (Neill, 2016). Sin un tratamiento antimicrobiano eficaz, la capacidad para llevar a cabo con éxito tanto los procedimientos médicos como el tratamiento de infecciones comunes podría volver a estar en peligro (Neill, 2016).

La resistencia antimicrobiana es la capacidad de un microorganismo (*e.g.*, bacterias, hongos, virus y parásitos) a sobrevivir a la exposición de un fármaco antimicrobiano (*e.g.,* antibióticos, antifúngicos, antivirales y antimaláricos) que en el pasado tenía la capacidad de matarlo o inhibir su crecimiento. Esta definición de resistencia antimicrobiana es referida como resistencia adquirida (Munita y Arias, 2016). Los

microorganismos que desarrollan resistencia antimicrobiana se denominan “superbacterias” (*superbugs*), por ejemplo: el Staphylococcus aureus resistente a meticilina y los enterococos resistentes a la vancomicina (Hampton, 2013). Un microorganismo puede adquirir la resistencia de dos maneras: la primera es por mutaciones, es decir, a través nuevos cambios genético asociados con el mecanismo de acción del antimicrobiano atacante; la segunda por transferencia horizontal, es decir a través de la adquisición de ADN de un microorganismo que ya es resistente (Hampton, 2013).

1.2. Péptidos antimicrobianos: una plantilla para la nueva generación de antimicrobianos.

Una posible solución para abordar el problema de la resistencia a los antimicrobianos es fomentar la investigación y el desarrollo de productos antimicrobianos nuevos y eficaces (Tacconelli *et al.*, 2018). Los péptidos antimicrobianos (AMPs) son moléculas cortas naturales que se encuentran presentes en la mayoría de los organismos vivos como parte del sistema de defensa innato con una amplia actividad para matar directamente bacterias, levaduras, hongos, virus e incluso células cancerosas (Zhang y Gallo, 2016). Desde el descubrimiento del primer péptido antimicrobiano hace más de 30 años (Zasloff, 1987), se ha considerado a estos como una alternativa a los antibióticos convencionales.

Los AMPs tienen propiedades terapéuticas deseables, como la actividad antimicrobiana directa y la baja susceptibilidad a la resistencia de los antimicrobianos. Además, se ha probado *in vitro* la eficacia que tiene en contra de patógenos microbianos multirresistentes (Cherkasov *et al.*, 2008b). Actualmente existen AMPs aprobados por la agencia de drogas y alimentos (FDA, por sus siglas en inglés de *Food and Drug Administration*), algunos ejemplos incluyen el péptido catiónico polimixina B y el glicopéptido no catiónico vacomicina. Por un lado, la polimixina B es efectiva contra las bacterias multiresistente gramnegativas pseudomona aeruginosa, acinetobacter baumannii y klebsiella pneumoniae (Zavascki *et al.*, 2007). Por otro lado, la vacomicina es efectiva en contra de las bacterias grampositivas tal como el Staphylococcus aureus resistente a la meticilina (Zhang y Gallo, 2016).

A pesar de que en el mercado se encuentran algunos antibióticos basados en AMPs,

todavía existen muchos otros que han sido declinados por la FDA en alguna de las fases correspondientes a las pruebas clínicas (Fox, 2013). Los AMPs todavía presentan algunas desventajas en la producción, en las propiedades farmacocinéticas y en la eficacia, por ejemplo: el costo de producción de los AMPs es elevado comparado con fármacos basados en moléculas pequeñas, los AMPs son altamente susceptibles a la degradación por proteasas y pueden ser tóxicos para el organismo huésped (Mahlapuu *et al.*, 2016; Brogden, 2005). Estas desventajas que presentan los AMPs son retos en el área de diseño de nuevos AMPs para desarrollar estrategias para la producción de péptidos estables, seguros y eficientes (Fjell *et al.*, 2012; Brogden, 2005).

1.3. Motivación de los métodos computacionales para el descubrimiento y diseño de los péptidos antimicrobianos.

La industria farmacéutica sufre actualmente un fenómeno conocido como la Ley de Eroom (inversa de la Ley de Moore) que indica que por cada mil millones de dólares invertidos en investigación y desarrollo, el número de nuevos fármacos aprobados por la FDA se ha reducido durante cada 9 años a la mitad desde 1950, esto de acuerdo con las cifras presentadas por Scannell *et al.* (2012). Esta situación motiva a buscar nuevas estrategias de diseño, siendo una de estas el enfoque computacional.

El descubrimiento de nuevas secuencias de péptidos bioactivos, a partir del espacio sintéticamente accesible de péptidos, es costoso y consume mucho tiempo (Tucker *et al.*, 2018; Fjell *et al.*, 2012). En primer lugar, el conjunto de descriptores moleculares que mejor discrimina a los péptidos bioactivos de aquellos que no, es un problema todavía no resuelto del todo. Segundo, aún en el caso de conocer dicho conjunto de descriptores el tamaño del espacio de las secuencias peptídicas posibles de longitud N es notablemente grande (20^N) (Fjell *et al.*, 2012; Neme *et al.*, 2017; Tucker *et al.*, 2018). Tercero, la tecnología para la evaluación *in vitro* de la actividad de los péptidos se encuentra limitada, por ejemplo los enfoques actuales de síntesis química permiten cribar solo un número limitado de secuencias cortas y lineales a la vez, estos requieren de la química combinatoria y de la robótica para aumentar el número de péptidos que se pueden probar en el laboratorio, lo que resulta prohibitivo para la mayoría de los investigadores de la academia (Tucker *et al.*, 2018; Hilpert *et al.*, 2007). Cuarto, actualmente existe una falta de comprensión de la química de los péptidos y la activi-

dad antimicrobiana. También, existe una limitante en la capacidad para explorar AMPs más allá de las secuencias disponibles de forma natural, dejando sin explorar la mayor parte del espacio químico, en donde también pueden existir AMPs potenciales valiosos para el desarrollo terapéutico (Tucker *et al.*, 2018).

Por otra parte, con la llegada de las tecnologías de secuenciación masiva (NGS como se las conoce en inglés de *Next-Generation Sequencing*) es posible generar una gran cantidad de datos (*e.g.,* ADN, ARN o proteínas) provenientes de diferentes seres vivos, en donde se pueden encontrar, en forma no clasificada, péptidos antimicrobianos. Sin embargo, la gran cantidad de datos que estas tecnologías producen dificultan, en extremo, la identificación de nuevos AMPs.

1.4. Planteamiento del problema y objetivos

Para hacerle frente a los problemas anteriores, las técnicas computacionales son atractivas para complementar las técnicas tradicionales de síntesis y prueba de péptidos, esto debido a que proporcionan información útil, además de que pueden proveer nuevos péptidos y permitir un análisis experimental costo-efectivo (virtual) de los péptidos antes de su síntesis y prueba en laboratorio (Fjell *et al.*, 2012). Esto motiva el desarrollo de métodos computacionales que: i) sean capaces de determinar el conjunto de características relevantes para identificar una función biológica, ii) evalúen la calidad del péptido y iii) exploren eficientemente el espacio de los péptidos posibles.

Por lo anterior, el problema de identificación y diseño de péptidos antimicrobianos *in silico* lo podemos caracterizar como un problema de búsqueda y por lo tanto se tiene que considerar lo siguiente:

- ¿Cómo buscar, de manera eficiente, en el espacio de los posibles conjuntos de descriptores de péptidos?
- ¿Cómo evaluar la calidad del conjunto de descriptores seleccionados?
- ¿Cómo buscar, de manera eficiente, en el espacio de las posibles secuencias de péptidos?
- ¿Cómo evaluar la calidad de los péptidos seleccionados?

Para responder a estas preguntas que definen el proyecto de investigación, se plantea el siguiente objetivo.

1.4.1. Objetivo general

Diseñar e implementar heurísticas evolutivas y técnicas de aprendizaje de máquina para la identificación y diseño de péptidos antimicrobianos partiendo de un conjunto de péptidos bioactivos aislados y caracterizados de la naturaleza. Para alcanzar este objetivo se propone los siguientes objetivos específicos.

1.4.1.1. Objetivos específicos

- Establecer un conjunto de prueba de secuencias de péptidos constituido por casos positivos (péptidos antimicrobianos con evidencia experimental reportada) y casos negativos (no antimicrobianos).
- Establecer un universo de descriptores moleculares computables en péptidos.
- Determinar un conjunto de descriptores moleculares relevantes, partiendo del universo obtenido en el objetivo anterior, para la clasificación de la actividad antimicrobiana.
- Construir un clasificador para la actividad antimicrobiana del péptido mediante algoritmos de aprendizaje de máquina.
- Proponer e implementar un algoritmo de aprendizaje de máquina para el diseño de péptidos antimicrobianos teniendo como referencia un conjunto específico de péptidos bioactivos aislados de la naturaleza.

1.5. Contribuciones

Las principales contribuciones de este proyecto de investigación son las siguientes:

- Un nuevo algoritmo evolutivo multi-objetivo para la selección de descriptores moleculares para la correcta clasificación de péptidos antimicrobianos y no antimicrobianos.

- Un nuevo algoritmo evolutivo basado en el modelo de envoltura para la selección de descriptores moleculares para la clasificación de péptidos antimicrobianos.
- Un conjunto de descriptores moleculares para la discriminación de péptidos antimicrobianos de no antimicrobianos.
- Un clasificador de péptidos antimicrobianos basado en redes neuronales profundas con un alto porcentaje de precisión de clasificación.
- Un conjunto de secuencias candidatas con posible actividad antimicrobiana con un nivel bajo de similitud con los péptidos antimicrobianos conocidos.

1.6. Organización de la tesis

El presente trabajo está organizado de la siguiente manera:

En el Capítulo 2 se exponen los conceptos biológicos básicos y que son necesarios para el entendimiento de este trabajo de investigación. Por otra parte, se abordan los conceptos computacionales para la comprensión del problema tratado en este trabajo. De igual modo se expone el trabajo previo relevante en la clasificación y diseño de AMPs.

En el Capítulo 3 se presenta la recopilación de los descriptores moleculares que se consideran en esta investigación para representar los péptidos en el espacio químico.

En el Capítulo 4 se presenta un método novedoso para seleccionar automáticamente una representación en los péptidos, basada en descriptores moleculares, que realiza eficientemente la clasificación de la actividad antimicrobiana en su forma binaria (*i.e.,* AMP y noAMP).

En el Capítulo 5 se presenta el problema de seleccionar automáticamente la representación adecuada de los péptidos. Para seleccionar la representación se propone una adaptación al exitoso método de ponderación de características propuesto por Paul y Das (2015).

En el Capítulo 6 se describe la metodología para generar nuevos AMPs con dos características: la primera, generar secuencias de AMPs no similares a un conjunto

conocido de AMPs; la segunda es que las secuencias generadas sean similares en el espacio químico.

Por último, en el Capítulo 7 se exponen las conclusiones a las que se llegó, así como algunas propuestas para la continuación de este trabajo de investigación.

Capítulo 2. Marco Teórico

2.1. Introducción a los péptidos antimicrobianos

Esta sección se enfoca a los aspectos básicos de los AMPs, qué son, cómo están formados, sus genes y sus mecanismos de acción.

2.1.1. Péptidos y los bloques que lo conforman

Los péptidos son moléculas formadas por diferentes bloques de construcción (*i.e.,* monómeros), denominados aminoácidos (aa), los cuales están unidos por enlaces peptídicos (ver Figura 1a). Los aminoácidos que forman parte del péptido se denominan residuos de aminoácidos ya que al unirse covalentemente se pierde una molécula de agua (ver Figura 1). Es importante señalar que los péptidos puede estar compuestos por cualquier número de residuos. Por ejemplo, un dipéptido contiene dos residuos, un tripéptido contiene 3, un oligopéptido contiene 3 a 10, y un polipéptido contiene más de 10 residuos (Hughes, 2013). Veinte aminoácidos son los que comúnmente forman parte de los péptidos naturales (ver Figura 2). Por lo tanto, si consideramos las posibles combinaciones de estos bloques para formar una cadena peptídica en particular, el número de péptidos que podemos formar es enorme. Por ejemplo, si consideramos péptidos de 10 aminoácidos de longitud, el espacio de las posibles secuencias es de $20^{10} = 1.024 \times 10^{13}$.

La estructura de los aminoácidos la podemos dividir en dos partes: la primera parte, común a los 20 aminoácidos, está constituida por dos grupos funcionales, un grupo carboxilo y un grupo amino, ambos unidos por un átomo de carbono, llamado carbono α; la segunda parte un tercer grupo funcional denominado cadena lateral, le da la propiedad y estructura química a cada uno de los 20 aminoácidos (ver Figura 1a). Por lo tanto, dependiendo de los residuos que formen al péptido es la forma estructural que este puede adoptar.

2.1.2. Péptidos antimicrobianos (AMPs)

De manera general, podemos definir un péptido antimicrobiano como aquel que mata o inhibe el crecimiento de microbios (*e.g.,* virus, hongos, bacterias y parásitos).

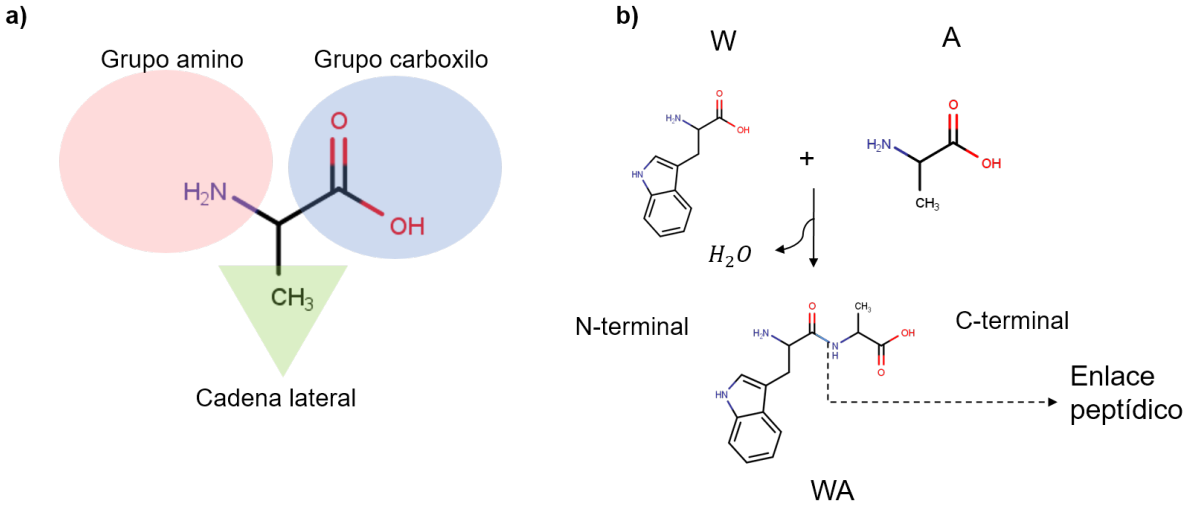


Figura 1. Estructura de los aminoácidos estándares y la formación de un dipéptido. a) La estructura de un aminoácido estándar está formado por un grupo amino (rosa) un grupo carboxilo (azul) unido a un átomo de carbono denominado carbono α. Además, contienen un grupo funcional que es el que distingue a cada uno de los 20 aminoácidos. b) En la parte superior se muestra dos distintos aminoácidos, un triptófano (W) y una alanina (A). En la parte inferior de la imagen se muestra un dipéptido, este es producido cuando el grupo amino de un aminoácido reacciona con el grupo carboxilo de otro para formar un enlace peptídico (Karp, 2009).

Los AMPs son parte de la respuesta inmune innata de varios organismos con diversas actividades antimicrobianas (Hancock y Diamond, 2000). Los AMPs varían en su longitud, estos van de 5 a 100 aminoácidos, y tiene un amplio espectro de organismos a los que atacan, que van desde virus hasta parásitos (Bahar y Ren, 2013).

2.1.2.1. Genética y la formación de AMPs

Los AMPs son codificados por simples genes como moléculas precursoras (*i.e.,* péptidos inactivos), las cuales incluyen una secuencia señal o prepéptido, seguida de una región propéptido (Hancock y Diamond, 2000; Taber, 2001). La región propéptido se procesa posteriormente para obtener el péptido antimicrobiano activo (ver Figura 3). A los eventos que ocurren para transformar el prepropéptido a su forma madura se le conoce como eventos proteolíticos (Taber, 2001). El mecanismo mediante el cual se transforma el prepropéptido a su forma madura es mediado por enzimas denominadas proteasas, que catalizan una reacción de proteólisis. La secuencia señal proporciona el objetivo hacia una estructura de membrana intracelular (*i.e.,* retículo endoplasmático en eucariontes o la membrana citoplasmática en procariontes), esta secuencia se separa tras una modificación postraduccional para activar el péptido maduro (Taber,

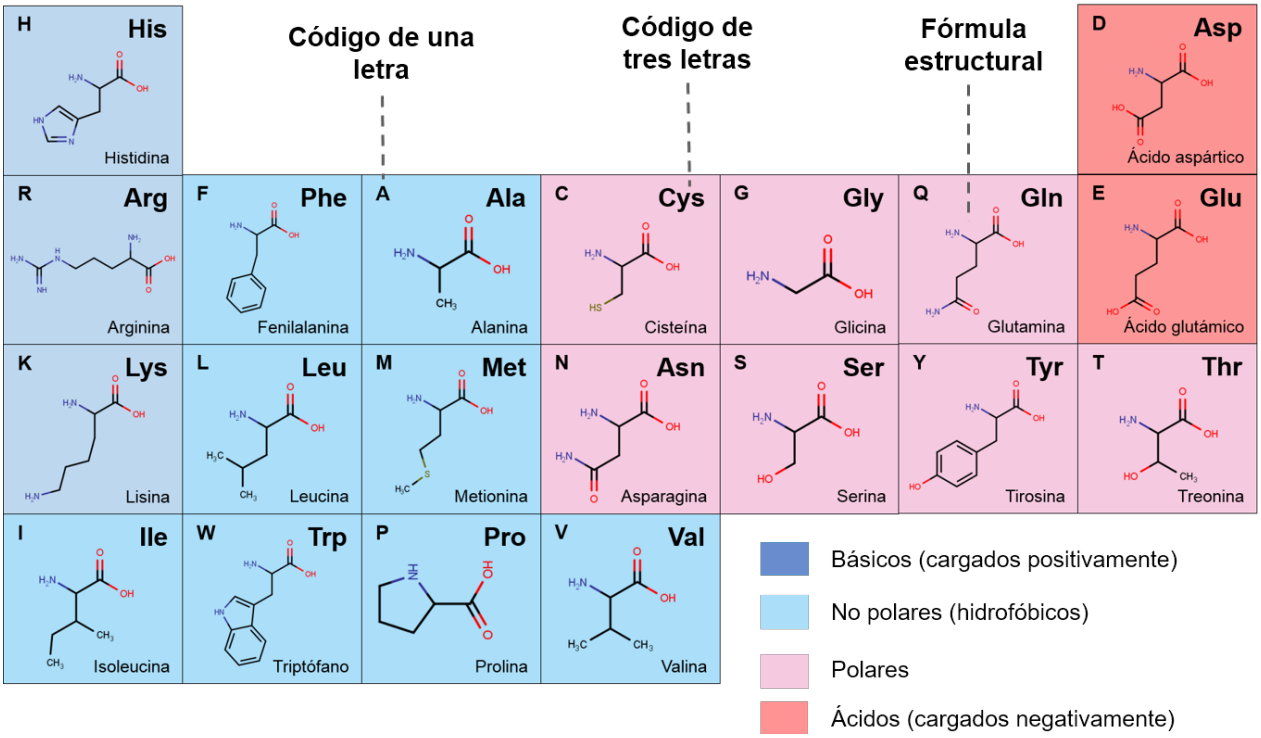


Figura 2. Aminoácidos estándares y sus diferentes representaciones. Los 20 aminoácidos se encuentran organizados por las propiedades y la estructura química de sus cadenas laterales.

2001).

La región propéptido, para algunos AMPs (e.g., defensinas), proporciona una neutralización del péptido maduro catiónico debido a su característica aniónica. Por lo tanto, la región propéptido inhibe la actividad del péptido maduro hasta su proceso de liberación. En la naturaleza existen diversos métodos por los cuales se puede remover la región propéptido para crear el péptido activo (Taber, 2001).

2.1.2.2. Mecanismos de acción

Actualmente se conocen al menos tres mecanismos de acción en AMPs para atacar las células de los microorganismos: el primero, por perturbación de la integridad de la membrana; el segundo, por la inhibición de la síntesis de proteínas, ADN y ARN; tercero, por la interacción con objetivos intracelulares (Bahar y Ren, 2013). De acuerdo con Bahar y Ren (2013), los péptidos sin importar su mecanismo de acción inicialmente requieren la interacción con la membrana celular para exhibir la actividad antimicrobiana.

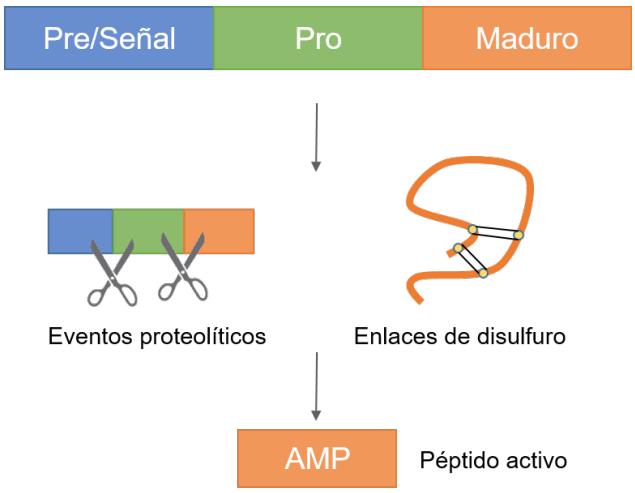


Figura 3. Representación de la estructura de un prepropéptido de un AMP, así como su proceso de ruptura para generar AMPs activos.

En general, los péptidos que se unen a la membrana basan su mecanismo de unión en las interacciones electrostáticas con la membrana celular cargada negativamente, esto es resultado de que muchos AMPs adoptan una estructura secundaria anfipática, *i.e.,* estos presentan ambas caras: catiónica (*i.e.,* hidrófila) e hidrofóbica. La parte cargada del péptido permite la interacción con la cabeza hidrófila del fosfolípido mientras que la parte hidrofóbica interactúa con el núcleo de la bicapas de fosfolípidos (Mahlapuu *et al.*, 2016). Después de la conformación del péptido inicia la introducción transversal del péptido en la bicapas de fosfolípidos mediante uno de los posibles mecanismos de acción que se muestran en las figuras 4a, 4b y 4c. Es importante resaltar que se necesita una concentración de péptidos mínima para que se lleven a cabo los mecanismos de acción que perturben a la célula del microorganismo objetivo (Yeaman y Yount, 2003).

Además de la ruptura de la membrana, otro mecanismo de acción de los AMPs es la penetración de la membrana con el objetivo de moverse hacia el citoplasma (ver Figura 4d), en donde se dirigen a procesos celulares clave tales como la síntesis de ADN, ARN y proteínas, el plegamiento de proteínas, la actividad enzimática y la síntesis de la pared celular (Mahlapuu *et al.*, 2016).

Para un tratamiento exhaustivo sobre los AMPs se refiere al lector a documentos recientes sobre el tema (Wang *et al.*, 2019; Matsuzaki, 2019).

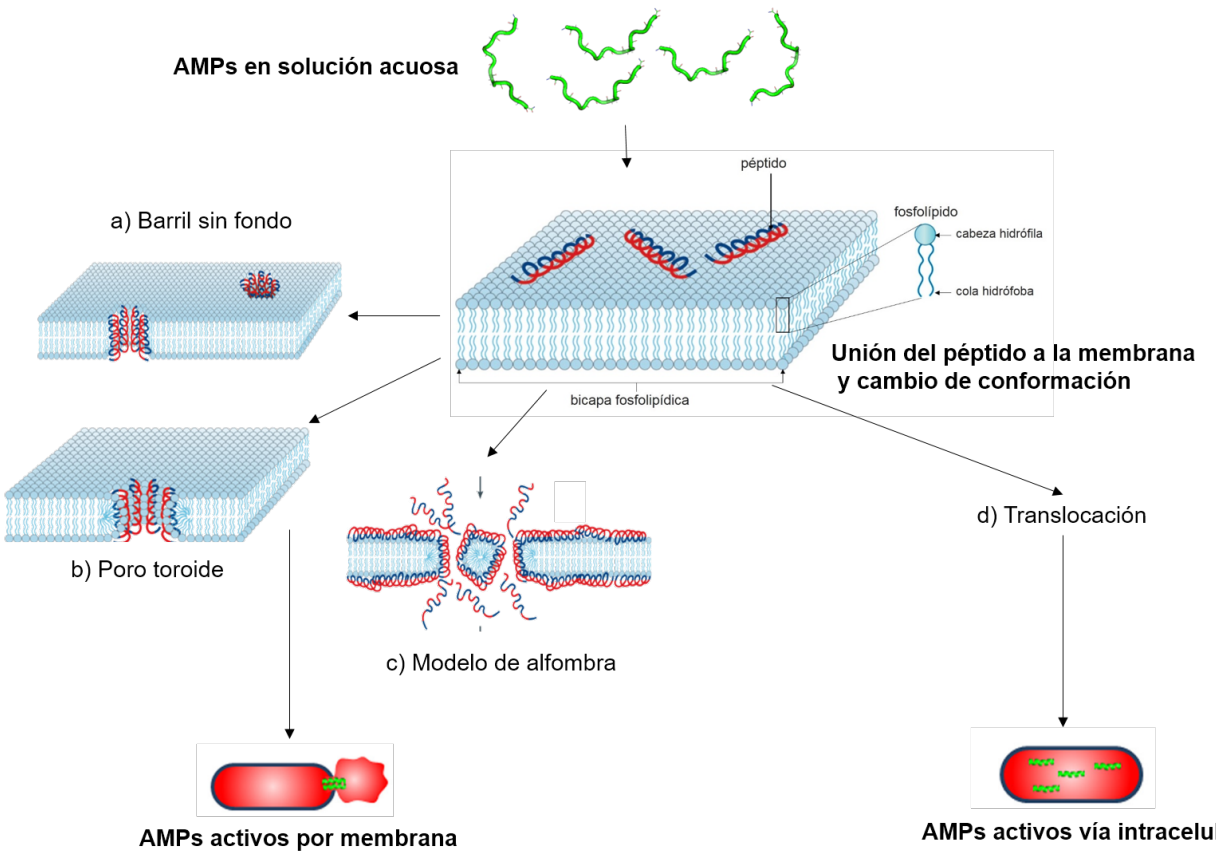


Figura 4. Representación esquemática de los mecanismos de acción de los AMPs. En general, los péptidos atacan a las células de los microorganismos por perturbación de la membrana o vía intracelular (Bahar y Ren, 2013). a) Barril sin fondo. b) Poro toroide. c) Modelo de alfombra (Brogden, 2005). d) Es otro mecanismo de acción en donde los AMPs interactúan con los objetivos dentro de las células.

2.2. Diseño y descubrimiento *in silico* de péptidos antimicrobianos

Para diseñar AMPs terapéuticos, es esencial identificar péptidos con las propiedades y la bioactividad deseada. Sin embargo, considerando el vasto espacio combinatorio de secuencias y las limitaciones prácticas para la evaluación *in vitro* de todas ellas (Tucker *et al.*, 2018), dicho diseño se torna una tarea extremadamente difícil (Tucker *et al.*, 2018; Fjell *et al.*, 2012). Las estrategias computacionales tales como el cribado virtual (VS por sus siglas en inglés de *Virtual Screening*) pudieran apoyar a superar esta limitante. VS tiene la ventaja de evaluar de manera automática grandes bibliotecas de péptidos con el objetivo de seleccionar AMPs potenciales antes de ser probados en el laboratorio experimental (*wet lab*). Además, VS pudiera encontrar secuencias de péptidos que no necesariamente ocurren en la naturaleza, *i.e.,* moléculas virtuales derivadas del diseño de moléculas nuevas asistido por computadora (Fjell *et al.*, 2012; Mannhold *et al.*, 2011).

Tres problemas principales necesitan ser resueltos en el diseño de AMPs asistido por computadora: primero, identificar un péptido como semilla que tenga la actividad deseada o crear unos desde cero; segundo, definir una función de evaluación para los péptidos; tercero, establecer una estrategia de búsqueda para explorar el vasto espacio de secuencias de péptidos (Fjell *et al.*, 2012).

La función de evaluación es uno de los aspectos más importantes en el diseño y por lo general se utiliza un modelo matemático que predice la actividad antimicrobiana del péptido en su forma discreta (*e.g.,* AMP o no AMP) o continua (mínima concentración inhibitoria, MIC). En este contexto, la relación cuantitativa de estructura-actividad (QSAR por sus siglas en inglés de *Quantitative Structure–Activity Relationship*) es de gran importancia para la generación de modelos de clasificación que estimen la actividad antimicrobiana de un péptido. El modelado QSAR define la relación matemática entre las propiedades fisicoquímicas de los péptidos (*i.e.,* descriptores moleculares) y su actividad biológica (Jenssen, 2011) para clasificar la actividad de nuevos péptidos. Los enfoques de aprendizaje automático estiman esta relación matemática a partir de un conjunto de péptidos con actividades conocidas. Es importante señalar que el rendimiento del modelo depende de la selección adecuada de los descriptores moleculares, ya que éstos definen el espacio químico en el que se proyecta cada péptido. La selección de descriptores moleculares que discriminen entre los distintos tipos de

péptido es una meta difícil de alcanzar, debido principalmente al gran número de descriptores moleculares que pueden ser calculados en los péptidos y a sus complejas interrelaciones.

Por otra parte, explorar el vasto espacio de las secuencias de péptidos no es una tarea fácil considerando el tamaño del mismo (*i.e.,* 20^n, donde n es la longitud del péptido). De manera interesante, incluso si se lograra tener un método que estimara con exactitud del 100 % la actividad antimicrobiana de los péptidos, la exploración de forma exhaustiva de las secuencias no sería factible. Por ejemplo, considere explorar todo el espacio de las secuencias de péptidos de longitud 10, en donde por cada secuencia la función de evaluación tarda un segundo en determinar si esta tiene o no la actividad, entonces el tiempo total de evaluación de todas las secuencias es de 20^{10} segundos, lo que es igual a 3247.08 siglos. Por lo anterior, es necesario establecer métodos eficientes de búsqueda para encontrar péptidos activos.

En las siguientes subsecciones se describen aspectos importantes tanto de la evaluación de la actividad antimicrobiana de los péptidos, como también de la búsqueda eficiente de AMPs en el espacio de las secuencias de péptidos.

2.2.1. Construcción del modelo para la predicción de la actividad antimicrobiana

En esta sección se explica cómo se construyen los modelos de predicción y qué medidas de calidad se usan para evaluar los modelos construidos.

2.2.1.1. Representación de los péptidos: cálculo de descriptores

La diversidad de formas y características de los AMPs conocidos hace que sea difícil crear modelos capaces de predecir la actividad antimicrobiana con un buen desempeño basándose sólo en la similitud de sus secuencias (Lira *et al.*, 2013). Es por esta razón que la mayoría de las propuestas para predecir AMPs utilizan una representación alterna basada en propiedades físicas y químicas de la estructura del péptido. Una de las principales diferencias entre la representación basada en la secuencia de aminoácidos y la de características es la longitud de la representación, la basada en

secuencia es variable mientras que la basada en características es de longitud fija. En la representación basada en características, cada péptido es representado por un vector de m dimensiones, $\mathbf{x^i} = [x_1^i, \ldots, x_m^i]$, utilizando descriptores moleculares.

Los descriptores moleculares son el resultado de un procedimiento lógico y matemático que transforma la información química del péptido en un número útil (Todeschini y Consonni, 2008). Los descriptores moleculares para estudios de péptidos antimicrobianos son clasificados en dos categorías dependiendo de cómo se obtuvieron: descriptores empíricos, se obtienen a partir de información medida en ensayos biológicos; descriptores calculados o basados en la estructura, son descriptores moleculares teóricos calculados a partir de una representación molecular (Hilpert *et al.*, 2008). Los descriptores moleculares basados en la estructura son transformados a una representación numérica mediante un procedimiento de cómputo. Los descriptores se clasifican en diferentes niveles de dimensión, esto dependiendo de la estructura molecular que se necesite, los niveles van desde la dimensión cero hasta la cuatro (Helguera *et al.*, 2008). Los descriptores 0D son propiedades moleculares muy simples (*e.g.,* masa molecular y conteo de átomos), estos dependen únicamente de la composición molecular del péptido. Los descriptores 1D codifican información sobre fragmentos estructurales moleculares (*e.g.,* distancia entre dos residuos de cisteínas, momento hidrofóbico). Los descriptores 2D son también conocidos como descriptores topológicos, y nos dan información contenida en un grafo molecular (por ejemplo, el índice de Wiener). Además, los descriptores 3D capturan la geometría molecular, estereoquímica y las propiedades de la superficie (Jenssen, 2011). El último tipo de descriptores es el 4D, los descriptores en esta clase capturan información relacionada a la interacción entre moléculas. Es importante señalar que, con una estructura molecular necesaria para el cálculo de descriptores moleculares de una dimensión dada, es posible calcular los descriptores de dimensiones menores a este. Además, entre mayor sea la dimensión del descriptor, mayor es la cantidad de información que se puede obtener (ver Figura 5).

En AMPs, los descriptores basados en estructuras 3D son difícil de calcular debido a que este tipo de estructura no está disponible para la mayoría de los péptidos en las bases de datos actuales.

Actualmente existen tanto software comerciales como gratuitos que ofrecen el

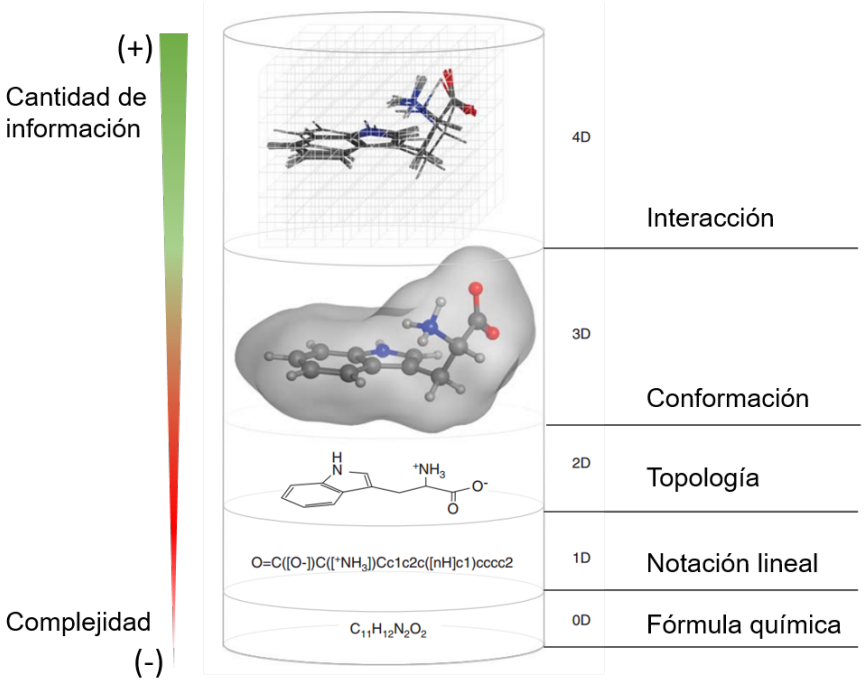


Figura 5. Descriptores moleculares clasificados de acuerdo con su dimensión. Los descriptores se encuentran organizados de los menos complejos a los más complejos (de abajo hacia arriba), esto es de 0D a 4D (Jenssen, 2011).

cálculo de miles de descriptores sobre moléculas naturales y muchos de ellos se adaptan de acuerdo con el tipo de molécula (Fjell *et al.*, 2012).

2.2.1.2. Selección de características

Actualmente el número de descriptores medibles en los péptidos se encuentra en el orden de los miles, por lo que elegir los descriptores moleculares que capturen las propiedades relevantes de los AMPs se torna una tarea difícil. Las principales razones de esta dificultad son: primero, no se conoce una regla determinista que gobierne la elección de los descriptores; segundo, explorar el espacio de todos los posibles subconjuntos de descriptores no es factible, ya que el espacio de búsqueda es de tamaño $2^m - 1$ (donde m es el número de descriptores moleculares).

Para la selección de las características existen tres clases principales de métodos agrupados según la función de evaluación de las características: filtrado, envoltura e híbrido. En primer lugar, en los métodos de filtrado, la calidad de las características se evalúa a partir de los datos, ignorando el efecto de las características seleccionadas en el rendimiento del modelo de clasificación (Kohavi y John, 1997). Ejemplos de funciones de evaluación utilizadas en los métodos de filtrado son la distancia, la ganancia de

información y la medida de dependencia (Dash y Liu, 1997). En segundo lugar, los métodos de envoltura incorporan el rendimiento del clasificador (*e.g.,* la tasa de error, la precisión) para evaluar la calidad de las características seleccionadas (Kohavi y John, 1997). Finalmente, los métodos híbridos combinan ambos, el filtrado y los métodos de envoltura (Huang *et al.*, 2007).

Los métodos de envoltura generalmente superan a los métodos de filtrado, principalmente porque la selección de las características óptimas está sesgada hacia el efecto de estas características en el desempeño del clasificador. Además, los métodos de envoltura tienen un alto costo computacional debido a que requieren inducir y probar un clasificador por cada subconjunto de características a evaluar. Por el contrario, dado que los métodos de filtrado son independientes del algoritmo de clasificación, estos pueden ser implementados eficientemente (Kohavi y John, 1997). Además, los métodos de filtrado pueden mejorar su rendimiento utilizando medidas de evaluación para un algoritmo de clasificación específico (Kohavi y John, 1997). Por ejemplo, la distancia dentro de la clase podría ser apropiada para los algoritmos de aprendizaje basados en ejemplos, mientras que la ganancia de información podría serlo para los clasificadores basados en árboles de decisión.

2.2.1.3. Construcción del modelo

Una vez que los descriptores moleculares se calculan para cada uno de los péptidos, el siguiente paso es usar estas medidas para predecir otra propiedad de interés (*e.g.*, actividad antimicrobiana, toxicidad). Los modelos para predecir AMPs se organizan de acuerdo con el tipo de variable de salida en dos categorías: modelos de regresión, sirven para predecir la actividad del péptido, utilizando la actividad como una variable continua (e.g., predecir la mínima concentración inhibitoria (MIC)); clasificación, sirve para predecir la actividad de un péptido como activa o inactiva (Hilpert *et al.*, 2008), es decir, la variable de salida es discreta.

Ejemplos de algoritmos de aprendizaje de máquina incluyen máquina de soporte vectorial (SVM) (Fan *et al.*, 2008), bosques aleatorios (RF) (Breiman, 2001), los *k* vecinos más cercano (k-NN) (Aha y Kibler, 1991), perceptrón multicapa (MLP) y algoritmo c4.5 (Quinlan, 1993).

2.2.1.4. Medidas de desempeño

Con el objetivo de evaluar la eficacia de los modelos de clasificación consideramos diversas medidas de evaluación, estas han sido utilizadas para medir el desempeño de los modelos de clasificación de la actividad antimicrobiana del estado del arte (ver Subsección 2.4). Las medidas que se consideran para este trabajo de tesis son: exactitud (Acc), coeficiente de correlación de Matthews (MCC), precisión (Prec), especificidad (Spec), sensibilidad (Sens), exactitud balanceada (BalACC) y el área bajo la curva ROC (AUC por sus siglas en inglés). Las medidas de desempeño se utilizan tanto en el conjunto de entrenamiento como el de pruebas, esto sin importar la técnica de remuestreo (*e.g.,* validación cruzada y *bootstrapping*) que se utilice para la partición del conjunto de datos.

Cada una de las medidas de desempeño anteriormente mencionadas las podemos categorizar en dos tipos: desempeño específico de la clase y desempeño general. Por ejemplo, las medidas de Sens y Spec son medidas para la clase positiva y negativa, respectivamente. Mientras que las medidas Acc, MCC, BalACC y AUC nos muestran un panorama general de desempeño (James *et al.*, 2013a). Es importante notar que la medida de ACC no muestra un panorama general de desempeño cuando el conjunto de datos está desbalanceado. Sin pérdida de generalidad, las secuencias de la clase AMP son consideradas como positivas y las de la clase no AMP como negativas, respectivamente; entonces las medidas pueden ser formalmente definidas como sigue:

- Acc (Baldi *et al.*, 2000):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- MCC (Baldi *et al.*, 2000):

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}} \tag{2}$$

- Prec:

$$Prec = \frac{TP}{TP + FP} \tag{3}$$

- Sens:

$$Sens = \frac{TP}{TP + FN} \tag{4}$$

- BalAcc (Gabere y Noble, 2017):

$$BalAcc = \frac{1}{2}\left(\frac{TP}{TP + FN}\right) + \frac{1}{2}\left(\frac{TN}{TN + FP}\right) \tag{5}$$

donde TP, TN, FP y FN es el número de verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, respectivamente.

2.3. Trabajo previo relevante

A continuación, describimos los distintos conjuntos de datos usados por métodos del estado del arte para el problema de clasificación de AMPs.

2.3.1. Conjuntos de datos de referencia para el problema de la clasificación de AMPs

En la literatura es difícil encontrar un único conjunto de referencia con el cual se pruebe el desempeño de un nuevo modelo de predicción de secuencias antimicrobianas. Contrario a esto, es común que los autores presenten su propio conjunto de péptidos para probar la exactitud de sus métodos. Por lo tanto, para este trabajo de tesis, seleccionamos diversos conjuntos de referencia tomando en cuenta su disponibilidad y el desempeño que obtuvo el predictor de AMPs en el cual se utilizó.

A continuación, se describen los conjuntos de referencia que se utilizaron en este trabajo de tesis.

2.3.1.1. Conjuntos de referencia propuestos por Gabere y Noble (2017)

Gabere y Noble (2017) recolectaron dos conjuntos de referencia compuestos cada uno por secuencias de péptidos antimicrobianas (*i.e.,* conjunto positivo) y no antimicrobianos (*i.e.,* conjunto negativo). Además, los conjuntos positivos se dividen en tres categorías: antimicrobianos, antibacterianos y bacteriocinas.

Para formar los conjuntos positivos utilizan dos bases de datos de propósito general de AMPs. La primera base de datos es DAMPD (Seshadri Sundararajan *et al.*, 2011), de la cual se obtienen 1232 secuencias. Estas secuencias presentan la región madura y el propéptido. La segunda base de datos es APD3 (Wang *et al.*, 2016), de donde se obtienen 2338 secuencias con región madura solamente (*i.e.,* péptidos de longitud menor a 100 aa). Después, para evitar tener secuencias redundantes, Gabere y Noble (2017) utilizan CD-HIT (Huang *et al.*, 2010) para agrupar las secuencias que tienen un porcentaje de identidad de al menos el 90 % y tomar un representante por cada agrupamiento. Es importante notar que el representante en CD-HIT es el péptido con la cadena de aminoácidos más larga. Como resultado, se obtiene los conjuntos positivos con las siguientes secuencias: 547 para DAMPD y 1713 para APD3. En particular, de las 547 secuencias 313 corresponden a péptidos antibacterianos, 31 a bacteriocinas (subconjunto de los antibacterianos) y 234 otros AMPs. Por otra parte, de las 1713 secuencias de APD3, 1446 corresponden a péptidos antibacterianos, 154 bacteriocinas y 113 otros AMPs.

Los conjuntos negativos, *i.e.,* no AMPs, son seleccionados aleatoriamente de una supercadena generada de la concatenación de proteínas recuperadas de UniProt (Consortium *et al.*, 2018). Un hecho interesante es que ninguna de las proteínas, recuperadas para generar los no AMPs, ha sido anotada como antimicrobiana, y algunas de ellas son intracelulares. Por cada secuencia en el conjunto de AMPs se toman aleatoriamente seis secuencias de la misma longitud provenientes de la supercadena (Gabere y Noble, 2017) (ver Figura 6).

De aquí en adelante nombramos estos los conjuntos de referencia siguiendo con dos criterios: i) las siglas de la base de datos de donde se recuperaron los AMPs; ii) la actividad biológica anotada. En cuanto a su base de datos, nombramos a los conjuntos de datos como DAMPD y APD3, respectivamente. En cuanto a la actividad anotada, denominamos a los conjuntos de datos AMP, ANTIBACTERIAL y BACTERIOCIN. Los AMP son péptidos que tienen actividad antimicrobiana. Por otro lado, ANTIBACTERIAL es un subconjunto propio de AMP que son capaces de inhibir o matar bacterias. Además, BACTERIOCIN es un subconjunto propio de ANTIBACTERIAL, que contiene péptidos producidos por bacterias para inhibir o matar bacterias (estos péptidos se denominan bacteriocinas, se refiere al lector interesado a la base de datos Wang *et al.* (2010) para

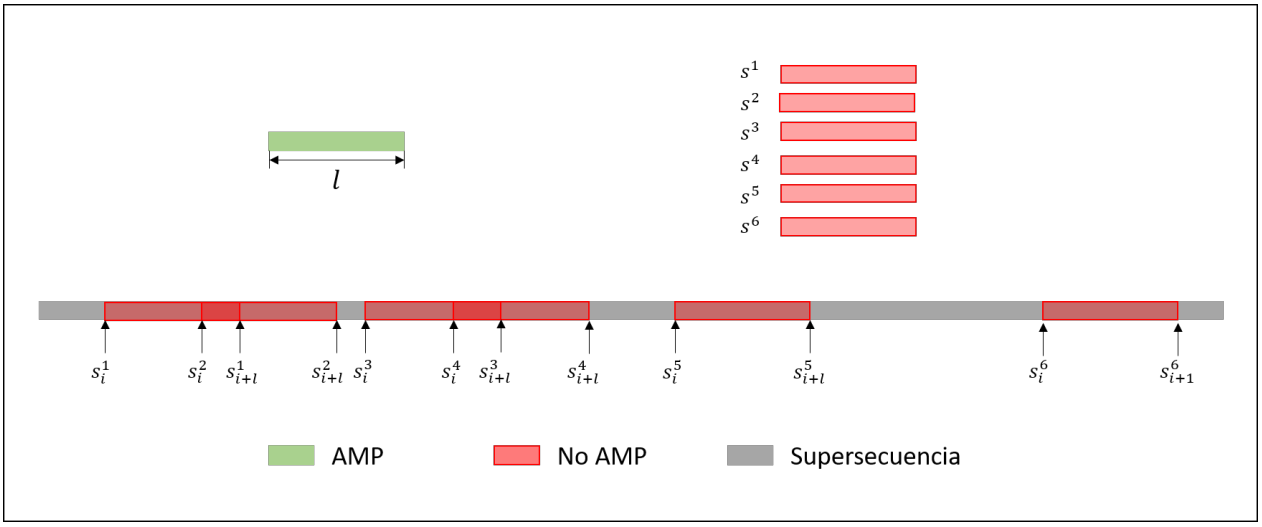


Figura 6. Representación gráfica de la estrategia propuesta por Gabere y Noble (2017) para crear el conjunto de péptidos no antimicrobianos.

más información sobre la denominación y clasificación de los péptidos).

2.3.1.2. Conjunto de referencia DAT1 propuesto por Fernandes *et al.* (2012)

El conjunto de referencia propuesto por Fernandes *et al.* (2012) contiene un menor número de secuencias: 115 AMPs y 116 no-AMPs (potenciales). Con respecto a las secuencias de AMPs, estas son experimentalmente validadas y tienen una longitud entre 10 y 100 residuos. Además, los péptidos de este conjunto se obtuvieron de la base de datos de péptidos antimicrobianos (APD) (Wang *et al.*, 2009). Por otra parte, para obtener secuencias para el conjunto de casos negativos, Fernandes *et al.* (2012) propusieron una metodología para generarlas ya que no existe una base de datos especializada en no AMPs. Estas secuencias, se recuperaron de la base de datos PDB (Berman *et al.*, 2000), y tienen una longitud entre 10 y 100 residuos. Enseguida, de las secuencias recuperadas del PDB, se eliminaron las secuencias que fueron predichas como de membrana o extracelulares, para este propósito utilizaron el servidor *Phobius* (Käll *et al.*, 2007). Posteriormente, con el fin de eliminar secuencias redundantes utilizaron CD-HIT (Huang *et al.*, 2010) con un porcentaje de identidad del 50 % para realizar los agrupamientos y después seleccionar la secuencia más larga por agrupamiento como representante.

Por último, en este trabajo al conjunto propuesto por Fernandes *et al.* (2012) lo nombramos DAT1.

2.3.1.3. Conjunto de referencia DAT2 propuesto por Thomas *et al.* (2009); Waghu *et al.* (2014)

El conjunto de referencia propuesto por Thomas *et al.* (2009) y Waghu *et al.* (2014) está constituido por 3010 AMPs y 4011 secuencias de no AMPs.

El conjunto de AMPs fue obtenido de la base de datos CAMP (Thomas *et al.*, 2009), en donde solo se recuperaron las secuencias patentadas y experimentalmente validadas. En este conjunto se removieron las secuencias que contienen aminoácidos no identificados (*i.e.,* contiene la letra 'X'), así como también, se eliminaron las secuencias redundantes utilizando el programa CD-HIT (Huang *et al.*, 2010) con un porcentaje de identidad de 100 %. Es importante señalar que en CD-HIT, la secuencia más larga es la que se mantiene como representante de cada conglomerado.

Por otra parte, el conjunto negativo se obtuvo de cuatro distintas formas: en la primera se obtuvo de secuencias experimentalmente validadas que no tiene la actividad antimicrobiana (25 secuencias). La segunda forma fue buscando en la base de datos UniProt (Consortium *et al.*, 2018) secuencias sin la anotación antimicrobiana y que tuvieran la etiqueta de proteínas no secretoras. Lo anterior, debido a que los AMPs presentan estas características. Cómo resultado se obtuvieron 2413 secuencias. La tercera forma fue a través de la generación aleatoria de secuencias (1200 secuencias). La cuarta forma fue seleccionar al azar 1200 secuencias, sin la anotación de antimicrobiana, de la base de datos UniProt (Consortium *et al.*, 2018). De forma similar que con el conjunto positivo, se utilizó el programa CD-HIT para eliminar las secuencias no redundantes con un porcentaje de identidad de 90 % (Thomas *et al.*, 2009).

En total el conjunto negativo es 1.5 veces más grande que el conjunto positivo. Este conjunto de referencia fue aleatoriamente divido en dos subconjuntos: el conjunto de entrenamiento compuesto por el 70 % del total de las secuencias y el conjunto de prueba compuesto por el resto de las secuencias. Por último, en este trabajo nombramos al conjunto de referencias propuesto por Thomas *et al.* (2009) y Huang *et al.* (2007) como DAT2.

2.3.1.4. Conjunto de referencia DAT3 propuesto por Xiao *et al.* (2013)

El conjunto de referencia propuesto por Xiao *et al.* (2013) está constituido por 1388 secuencias de AMPs y 1440 no AMPs (el conjunto de referencia fue obtenido de la información suplementaria S1 disponible en (Xiao *et al.*, 2013)).

El conjunto positivo está constituido a su vez por 1274 péptidos antibacterianos, 101 péptidos antitumorales/anticancer y 489 péptidos antifúngicos. En general, las secuencias de AMPs fueron obtenidas de la base de datos APD (Wang *et al.*, 2009, 2016) y se eliminó el sesgo de homología y redundancia con el programa CD-HIT (Huang *et al.*, 2010) con un porcentaje de identidad de 40 %. En CD-HIT, se toma como representante la secuencia de longitud más larga por conglomerado.

Por otra parte, el conjunto negativo se obtuvo de fragmentos y secuencias de proteínas de la base de datos UniProt (2012_08) (Consortium *et al.*, 2018), estas secuencias tienen longitud entre 5 y 100 aminoácidos. Del conjunto negativo se eliminaron secuencias con la anotación de antimicrobiano, antibiótico y fungicida, así como también, se eliminaron las secuencias con aminoácidos no estándares. De manera similar que en el conjunto positivo, con el fin de reducir el sesgo de homología y redundancia se utilizó CD-HIT con el mismo porcentaje de identidad.

2.4. Clasificación de la actividad antimicrobiana

La investigación computacional en péptidos antimicrobianos se ha centrado en la clasificación de la actividad antimicrobiana de las secuencias de péptidos. En este sentido, los métodos de clasificación, enfocados en AMPs, se pueden dividir en tres tipos: clasificación binaria de la actividad antimicrobiana, clasificación multiclase y clasificación multietiqueta.

Primero, la clasificación binaria de la actividad antimicrobiana, se refiere a determinar si el péptido tiene o no la actividad antimicrobiana (esta clasificación trata de responder la siguiente pregunta: ¿Es el péptido antimicrobiano? Sí o no). Para crear modelos que solucionen el problema de clasificación binaria se han propuesto varios trabajos, estos utilizan principalmente los siguientes algoritmos de aprendizaje de máquina: redes neuronales artificiales (ANN), sistema de inferencia neuro-adaptable (AN-

FIS), regresión logística, análisis de discriminantes (DA) (ver Tabla 1 para una referencia a los distintos métodos). En general, los algoritmos propuestos permiten generar modelos con una exactitud de clasificación de hasta el 96 %.

El segundo tipo es la clasificación multiclase, la cual tiene como objetivo determinar qué tipo de actividad tiene el péptido, esto por lo general se utiliza a *posteriori* de clasificar como antimicrobiano el péptido. Los distintos tipos de actividades son: antibacterial, anticancer/antitumoral, antifúngico, anti-VIH y antiviral (Xiao *et al.*, 2013). Además del tipo de actividad, existen otros enfoques en donde se determina la potencia de la actividad antimicrobiana en una de las siguientes categorías: ninguna, baja, media y alta (Lira *et al.*, 2013). Es importante notar que en la clasificación multiclase la asignación de la actividad es mutuamente excluyente, esto es que solo se puede asignar una etiqueta dada una secuencia. Para este problema de clasificación, Lira *et al.* (2013) presentan un modelo de árbol de decisión creado por el algoritmo J48 y reportan una exactitud del 70 %.

El tercer tipo es la clasificación multietiqueta, esta es similar a la clasificación multiclase, en donde la principal diferencia es que un péptido puede tener asignado más de una etiqueta de los distintos tipos de actividad antimicrobiana (Xiao *et al.*, 2013; Lin y Xu, 2016). En la literatura se han reportado AMPs que exhiben más de una actividad antimicrobiana, por ejemplo, el péptido Aurein 1.2 (Rozek *et al.*, 2000) experimentalmente ha mostrado atacar a bacterias Gram positivas, Gram negativas, virus y hongos. Los algoritmos que se han utilizado para crear modelos de clasificación multiqueta en AMPs son dos: el primero es *k*-vecinos más cercanos difuso (FKNN) (Xiao *et al.*, 2013); el segundo es bosques aleatorios (Lin y Xu, 2016). Los modelos de clasificación multietiqueta alcanzan una exactitud de hasta el 68 %.

En resumen, los algoritmos propuestos permiten generar modelos con una exactitud predictiva de hasta el 96 % para el caso de clasificación binaria y para la clasificación multietiqueta una ACC de hasta el 68 %. Sin embargo, los resultados de rendimiento de los métodos de predicción de los AMPs *in silico* no son comparables dado las discrepancias en el conjunto de datos utilizados tanto para entrenar como para probar los modelos de clasificación. Aunado a los diferentes métodos de estimación del rendimiento que se utilizan. Por otra parte, en lo que se refiere a la clasificación multiqueta y multiclase, el principal problema es el alto desbalance en el número de

Tabla 1. Trabajo relacionado en la clasificación binaria de AMPs.

Algoritmo de inducción	Referencia
Redes neuronales artificiales (ANN)	(Fjell *et al.*, 2009) (Thomas *et al.*, 2009) (Torrent *et al.*, 2011) (Waghu *et al.*, 2014)
Sistema de inferencia neuro-adaptable (ANFIS)	(Fernandes *et al.*, 2012)
Regresión logística (LR)	(Randou *et al.*, 2013) (Veltri *et al.*, 2017)
Bosques aleatorios (RF)	(Thomas *et al.*, 2009) (Waghu *et al.*, 2014)
Máquina de soporte vectorial con kernel polinomial (SVM)	(Thomas *et al.*, 2009) (Torrent *et al.*, 2011) (Waghu *et al.*, 2014)
Redes neuronales profundas (DNN)	(Hamid y Friedberg, 2018) (Veltri *et al.*, 2018)

secuencias de péptidos por actividad con la que se cuentan. Por ejemplo, considerando solo la base de datos APD3 (Wang *et al.*, 2016), el 80 % de las secuencias presentan la actividad antibacteriana.

2.4.1. Diseño *in silico* de AMPs

El diseño de nuevas secuencias de péptidos bioactivos, considerando el espacio de péptidos sintéticamente disponible, es costoso y consume mucho tiempo (Tucker *et al.*, 2018; Fjell *et al.*, 2012). Esto como consecuencia de que el espacio de las secuencias peptídicas es notablemente grande y la tecnología para su evaluación *in vitro* es limitada, así como también, existe una limitada comprensión entre la estructura química de los péptidos y la actividad antimicrobiana (Tucker *et al.*, 2018).

Los enfoques de diseño y descubrimiento basados en computadora han surgido para ayudar a la generación y evaluación sistemática de grandes bibliotecas de péptidos para seleccionar AMPs putativos con las propiedades deseadas antes de los ensayos biológicos. Trabajos anteriores han demostrado que los enfoques de diseño *in silico* son capaces de encontrar nuevas secuencias de péptidos a partir de secuencias naturales de AMP (Fjell *et al.*, 2012; Mannhold *et al.*, 2011; Porto *et al.*, 2018b; Fjell *et al.*, 2012; Cherkasov *et al.*, 2008b). Además, las secuencias diseñadas muestran una potencia comparable con los AMPs naturales conocidos (Fjell *et al.*, 2012; Fields *et al.*, 2018; Porto *et al.*, 2018b; Eliseev *et al.*, 2018). Los enfoques de diseño *in silico* para la generación de las secuencias utilizan modelos lingüísticos explícitos (Loose *et al.*, 2006;

Porto *et al.*, 2018a; Eliseev *et al.*, 2018), algoritmos genéticos guiados por propiedades físico-químicas (Beltran y Brizuela, 2016; Porto *et al.*, 2018b), redes neuronales artificiales (Cherkasov *et al.*, 2008b), programación genética (Veltri *et al.*, 2017), generación de secuencias aleatorias (Schneider *et al.*, 2017), estudios de plantillas basados en secuencias alteradas de aminoácidos de AMPs conocidos (Tossi *et al.*, 1997; Chen *et al.*, 2005; Robinson, 2011). Adicionalmente, enfoques recientes utilizan el aprendizaje profundo para producir modelos lingüísticos implícitos mediante el uso de redes neuronales recurrentes (RNNs) (Nagarajan *et al.*, 2018; Müller *et al.*, 2018).

2.4.2. Oportunidades de investigación en el diseño e identificación *in silico* de AMPs

En esta subsección se describen las oportunidades de investigación en el diseño e identificación de AMPs utilizando un enfoque *in silico*.

Primero, para generar un modelo QSAR para la identificación de AMPs existen dos aspectos cruciales: la elección del conjunto de descriptores que definen la característica de los péptidos de interés y la selección del algoritmo de aprendizaje de máquina para crear el modelo. La investigación computacional se ha centrado en el segundo aspecto, en el que se han propuesto varios algoritmos de aprendizaje por máquina (MLA) con este fin. Por el contrario, la selección de descriptores apropiados para la representación de péptidos ha recibido poca atención por parte de la comunidad científica, aunque es un aspecto esencial para determinar el rendimiento de los modelos predictivos, ya que esos descriptores definen el espacio químico en el que se proyecta cada péptido y, en consecuencia, la eficacia de la clasificación depende de ello. Además, actualmente se puede calcular un gran número de descriptores para los péptidos. En estudios anteriores, la selección de descriptores moleculares se ha hecho a menudo basándose en la intuición química o en las propiedades observadas que dan lugar a la actividad antimicrobiana (Fjell *et al.*, 2012; Torrent *et al.*, 2011). Por otro lado, estudios recientes emplean procedimientos de selección de características (descriptores) o métodos de filtrado que evalúan independientemente las características de acuerdo a un criterio dado y seleccionan las principales k características (Torrent *et al.*, 2011; Waghu *et al.*, 2014; Fernandes *et al.*, 2012). Sin embargo, la mayoría de estos enfoques se centraron en la relación e interacción de los descriptores por pares, mientras que

la actividad biológica podría depender de la relación de tres o más descriptores. Por lo tanto, se necesita un procedimiento de selección de características más exhaustivo para mejorar el rendimiento de los modelos de aprendizaje (Gabere y Noble, 2017).

Segundo, a pesar del rápido y sustancial progreso en el diseño *in silico* de AMPs, existen algunos inconvenientes relacionados con la estrategia de explorar el enorme espacio de las secuencias peptídicas. Por ejemplo, la mayoría de estos enfoques generan secuencias más cercanas a las secuencias de AMPs conocidas en la base de datos (Porto *et al.*, 2018b); por lo tanto, sólo se explora un área estrecha en todo el espacio molecular, lo que causa una escasa diversidad en los péptidos diseñados. La generación de secuencias diversas es significativa porque los péptidos candidatos pueden fallar en etapas posteriores del proceso de descubrimiento de fármacos (Benhenda, 2017). Por estas razones, deberían desarrollarse nuevos métodos de diseño *in silico* para la generación de péptidos antimicrobianos deseables con diversidad de secuencias internas y externas.

Para superar estas dificultades en este trabajo de tesis se proponen dos enfoques para la seleccionar automáticamente la representación de un péptido, basada en descriptores moleculares, que pueda eficientemente clasificar actividad antimicrobiana (ver capítulos 4 y 5). También, en este trabajo se propone una metodología para generar nuevos AMPs no similares en el espacio de secuencias y a su vez que sean similares en el espacio químico de un conjunto de entrada de AMPs, respectivamente (ver Capítulo 6).

Capítulo 3. Descriptores moleculares para péptidos antimicrobianos

Varios estudios han encontrado cinco propiedades principales relacionadas con la actividad antimicrobiana de los péptidos; éstas incluyen conformación, carga, carácter hidrofóbico y estructura secundaria (Kang *et al.*, 2017; Mahlapuu *et al.*, 2016; Fjell *et al.*, 2012; Wang *et al.*, 2010). En esta dirección, los descriptores moleculares han sido ampliamente aplicados para extraer estas propiedades en los péptidos de una manera cuantitativa. En este sentido, en este trabajo hemos recolectado e implementado los descriptores moleculares relacionados a estas cinco propiedades. A la librería encargada de calcular los descriptores moleculares, dado un conjunto de péptidos representados en secuencias de aminoácidos, la denominamos MODAMP (por sus siglas en inglés de MOlecular Descriptor for AntiMicrobial Peptides). A continuación, se describen los descriptores moleculares que se implementaron y los detalles para el cálculo de los mismos.

3.1. Materiales y métodos

En este estudio, se recopilaron un total de 118 descriptores moleculares, de los cuales se derivaron 268 valores. La cantidad de descriptores moleculares para cada propiedad fue: 74 en conformación, 10 de carga, 31 en carácter hidrofóbico, 5 en estructura secundaria y 2 en otras propiedades. Los descriptores moleculares incluidos en este trabajo han sido utilizados en estudios previos de péptidos antimicrobianos. En la Tabla 2 se muestra el resumen de todos los descriptores moleculares que recopilamos e implementamos. Estos descriptores se pueden calcular a partir de la secuencia peptídica. A continuación, se describe cada uno de los descriptores moleculares. Por otra parte, la lista de los 268 valores está disponible en el material complementario (archivo con nombre MODAMP_ListaDeDescriptores.xls).

3.1.1. Alfabetos reducidos: aminoácidos estándares y la agrupación por atributos

Para el cómputo de los descriptores moleculares utilizamos el alfabeto compuesto por los 20 aminoácidos estándar, representados en código de una letra, y 10 alfabetos

Tabla 2. Resumen de los 268 descriptores moleculares recopilados e implementados para la representación de los péptidos en el espacio químico. Estos descriptores se encuentran agrupados de acuerdo a la representación estructural necesaria para realizar el cómputo de los mismos.

Grupo	Propiedad*	Nombre	Núm. de descriptores moleculares	Núm. de valores generados	Referencia
0D	C	Composición estándar de los aminoácidos	1	20	(Gasteiger *et al.*, 2005)
	C	Composición reducida de los aminoácidos	10	41	(Dubchak *et al.*, 1995; Li *et al.*, 2006)
	O	Índice alifático	1	1	(Gasteiger *et al.*, 2005)
	Z	Carga neta y carga neta promedio	6	6	(Klein *et al.*, 1984)
	H	Promedio hidrofilicidad	2	2	(Klein *et al.*, 1984)
	H	Promedio de hidropatía (GRAVY)	1	1	(Gasteiger *et al.*, 2005)
	H	Promedio de hidrofobicidad	23	23	(Klein *et al.*, 1984)
	Z	Carga a diferentes valores de pH (pH con valores de 5, 7 y 9)	3	3	(Piotto *et al.*, 2012)
	H	Índice de Boman	1	1	(Boman, 2003)
	O	Masa molecular	1	1	(Gasteiger *et al.*, 2005)
	C	Número de aminoácidos	1	1	(Gasteiger *et al.*, 2005)
1D	S	Índice de inestabilidad	1	1	(Gasteiger *et al.*, 2005; Guruprasad *et al.*, 1990)
	C	Transición entre aminoácidos reducidos	10	21	(Dubchak *et al.*, 1995; Li *et al.*, 2006)
	C	Distribución de aminoácidos reducidos	50	105	(Dubchak *et al.*, 1995; Li *et al.*, 2006)
	C	Dipéptido	1	9	(Li *et al.*, 2006)
	C	Tripéptido	1	27	(Li *et al.*, 2006)
	H	Hidrofobicidad promedio máxima	1	1	(Eisenberg *et al.*, 1984)
	H	Momento hidrofóbico	3	3	(Eisenberg *et al.*, 1982)
	Z	Punto isoeléctrico	1	1	(Kozlowski, 2016; Gasteiger *et al.*, 2005)
		Total	**118**	**268**	

* Las letras corresponden a las siguientes propiedades: C=conformación; Z=Carga; H=carácter hidrofóbico; S=estructura secundaria; O= otros

reducidos. Un alfabeto reducido se produce de la agrupación de los aminoácidos de acuerdo a un atributo en particular, por ejemplo: carga, hidrofobicidad, polaridad. A continuación, definimos un alfabeto reducido de aminoácidos como sigue:

Alfabeto reducido de aminoácidos. Sea Σ el alfabeto compuesto por los 20 aminoácidos estándares representados en código de una letra y p un atributo de los aminoácidos. Definimos un alfabeto reducido de aminoácidos Σ_p como k subconjuntos de aminoácidos $C_1, \ldots, C_k$ sobre Σ, en donde cada aminoácido en Σ es asignado a uno de los k subconjuntos de acuerdo con el atributo p.

En la Tabla 3 se muestra los diez alfabetos reducidos de aminoácidos que utilizamos para el cálculo de los descriptores moleculares. La tabla muestra en cada renglón un atributo y los grupos que se contemplan, además de indicar a cuál grupo pertenece cada aminoácido.

De manera particular, los alfabetos reducidos se utilizan para calcular los siguientes descriptores moleculares: composición reducida de los aminoácidos, transición y distribución de aminoácidos reducidos, dipéptido y tripéptido.

Tabla 3. Alfabetos reducidos para los aminoácidos estándares organizados por atributo y división.

Atributo	Divisiones								Referencia
Hidrofobicidad	Hidrofóbico CFLMVWI	Hidrofóbico moderado AG	Polar moderado NQSTY	Polar HP	Muy polar DEKR				(Rose *et al.*, 1985)
BLOSUM50	Hidrofóbico leve P	Gpo. hidrofílico con alcohol ST	Hidrofóbico grande CLVIM	Hidrofóbico moderado AG	Cadena larga con carga + KR	Hidrófilo leve H	Cadena larga hidrofóbico FWY	Gpo. hidrofílico cargados/ polares DENQ	(Murphy *et al.*, 2000)
Similitud de conformación	Grupo I SCMEQKRL	Grupo III DN	Grupo IV HFYW	Grupo V VIT	Excluir A	Excluir G	Excluir P		(Pal y Chakrabarti, 2000)
Hidrofobicidad	Polar RKEDQN	Neutral GASTPHY	Hidrofóbico CLVIMFW						(Li *et al.*, 2006; Tomii y Kanehisa, 1996)
Volumen normalizado Van der Waals	Volumen intervalo 0-2.78 GASTCPD	Volumen intervalo 2.95-94 NVEQIL	Volumen intervalo 4.03-8.08 MHKFRYW						
Polaridad	Valor de Polaridad 4.9 - 6.2 LIFWCMVY	Valor de polaridad 8.0 - 9.2 PATGS	Valor de polaridad 10.4 - 13.0 HQRKNED						
Polarizabilidad	Valor de 0-1.08 GASDT	Valor de 0.128-120.186 CPNVEQIL	Valor de 10.4-13 KMHFRYW						
Carga	Positiva KR	Neutral ANCQGHIL MFPSTWYV	Negativa DE						
Estructura secundaria	Hélice EALMQKRH	β-lámina (strand) VIYCWFT	Desordenada (coil) GNPSD						
Accesibilidad al solvente	Enterrado ALFCGIVW	Expuesto RKQEND	Intermedio MPSTHY						

3.1.2. Composición, transición y distribución de aminoácidos reducidos

Los descriptores de composición (C), transición (T) y distribución (D) son útiles para describir la composición global de un atributo de aminoácido en un péptido dado, la frecuencia con la que la propiedad cambia a lo largo del péptido, así como el patrón de distribución de la propiedad en la secuencia (Dubchak *et al.*, 1995). A continuación definimos formalmente los descriptores y damos algunos ejemplos para facilitar el entendimiento.

Composición reducida de los aminoácidos C. Dado un alfabeto reducido de aminoácidos Σ_p y una secuencia de aminoácidos válida $S = s_1, \ldots, s_l$, donde cada aminoácido s_j está en Σ, entonces la composición reducida de los aminoácidos $C(C_i, S)$ para un subconjunto de aminoácidos C_i en Σ_p está definida como:

$$C(C_i, S) = \frac{N_{Ci}}{\sum_{j=1}^{k} N_{C_j}} \times 100, \tag{6}$$

donde N_{Ci} es el número de ocurrencias de C_i en S y $\sum_{j=1}^{k} N_{C_j}$ es la suma del número de ocurrencias de cada grupo del alfabeto reducido de aminoácidos en la secuencia S. Es importante notar que este descriptor genera el mismo número de valores que grupos existentes en el alfabeto reducido de aminoácidos.

En la Figura 7 se presenta un ejemplo del cómputo de la composición para el alfabeto reducido Σ_{carga}. En este ejemplo, la secuencia con código PDB 2FBU, contiene el 50 % de sus residuos con carga neutra.

Transición reducida de los aminoácidos T. Dado un alfabeto reducido de aminoácidos Σ_p y una secuencia de aminoácidos válida $S = s_1, \ldots, s_l$, donde cada aminoácido s_j está en Σ. La transición reducida de aminoácidos $T(C_i, C_j, S)$ se define como el porcentaje de la frecuencia con el cual el grupo C_i es seguido por otro grupo C_j y viceversa, donde C_i y C_j son grupos distintos. Esto es:

$$T(C_i, C_j, S) = \frac{N_{C_i - C_j} + N_{C_j - C_i}}{l} \times 100, \tag{7}$$

Índice	1	2	3	4	5	6	7	8	9	10	11	12
Secuencia	L	L	G	D	F	F	R	K	S	K	E	K
Grupo positivo							1	2		3		4
Grupo neutral	1	2	3		4	5			6			
Grupo negativo				1							2	

Composición reducida de los aminoácidos (C)

$$C(Positivo, S) = 33.33 \qquad C(Neutral, S) = 50 \qquad C(Negativo, S) = 16.67$$

Figura 7. Ejemplo del descriptor molecular composición reducida de aminoácidos para el alfabeto Σ_{carga} y la secuencia del péptido antibacteriano (PDB 2FBU). Los colores de los residuos corresponden a la siguiente carga: verde para residuos con carga postiva; gris para los residuos con carga neutra; finalmente, rojo para los residuos con carga negativa.

Índice	1	2	3	4	5	6	7	8	9	10	11	12
Secuencia	L	L	G	D	F	F	R	K	S	K	E	K
Transición positivo-negative											1	
Transición negativo-positive												1
Transición positivo-neutral									1			
Transición neutral-positive						1				2		
Transición negativo-neutral				1								
Transición neutral-negativo			1									

Transición reducida de los aminoácidos (T)

$$\mathrm{T}(Positivo, Negativo, S) = \frac{1+1}{12} x\ 100 = 16.67,\ \mathrm{T}(Positivo, Neutral, S) = \frac{1+2}{12} x\ 100 = 25,\ \mathrm{T}(Neutral, Negativo, S) = \frac{1+1}{12} x\ 100 = 16.67$$

Figura 8. Ejemplo del descriptor molecular transición reducida de aminoácidos para el alfabeto Σ_{carga} y la secuencia del péptido antibacteriano (PDB 2FBU). Los colores de los residuos corresponden a la siguiente carga: verde para residuos con carga positiva; gris para los residuos con carga neutra; finalmente, rojo para los residuos con carga negativa.

donde $N_{C_i-C_j}$ y $N_{C_j-C_i}$ son el número de transiciones C_i-C_j y C_j-C_i, respectivamente. Por otro lado, l es la longitud de las secuencia S.

El número de valores que este descriptor genera es ${}_kC_2$, en donde k es el número de grupos en el alfabeto reducido. Por ejemplo, considere el alfabeto reducido de aminoácidos Σ_{carga} (ver Tabla 3, renglón ocho), este tiene $k = 3$ grupos, por lo tanto, el número de valores que producirá el descriptor de transición será de ${}_3C_2 = 3$.

En la Figura 8 se muestra un ejemplo del cómputo de la composición para el alfabeto reducido Σ_{carga}. En este ejemplo, la secuencia con código PDB 2FBU, contiene 25 % de transiciones entre el grupo positivo y negativo.

Distribución reducida de aminoácidos D. Este descriptor mide, para un atributo dado, la distribución de cada grupo a lo largo de la secuencia a través de la medición de cinco segmentos, dentro de los cuales se encuentra el primero, 25 %, 50 %, 75 % y el 100 % de los aminoácidos con una determinada propiedad (Dubchak *et al.*, 1995). Por ejemplo, con estos descriptores podemos responder preguntas del siguiente tipo: ¿Cuál es el porcentaje de la secuencia en donde se encuentra distribuido el primer aminoácido con carga positiva? Formalmente podemos definir la distribución reducida de aminoácidos como sigue:

Dado un alfabeto reducido de aminoácidos Σ_p, una secuencia de aminoácidos válida $S = s_1, \ldots, s_l$, en donde cada aminoácido s_j está en Σ, además de una propiedad de distribución d, la distribución reducida de aminoácidos $D(C_i, d, S)$ se obtiene como sigue:

$$D(C_i, d, S) = \frac{Pos(C_i, d, S)}{l} \times 100, \tag{8}$$

donde $Pos(C_i, d, S)$ es una función que devuelve el índice en S en donde se satisface la propiedad de distribución d para el grupo C_i. Además, l es la longitud de la secuencia S. El número de valores que este descriptor genera es $k \times 5$, donde k es el número de grupos en el alfabeto reducido.

En la Figura 9 se muestra un ejemplo del descriptor molecular de distribución reducida de aminoácidos para el grupo positivo (ver Tabla 3, renglón ocho) y la secuencia con código PDB 2FBU. En la figura se puede observar que los aminoácidos que pertenecen al grupo positivo se encuentran distribuidos en la segunda mitad de la secuencia.

3.1.3. *K*-meros en secuencias de péptidos

En el contexto de las secuencias de péptidos, un k-mero se refiere a todas las posibles subsecuencias de longitud k que se pueden obtener de una secuencia de longitud l, esto es igual a $l - k + 1$. Por otra parte, si consideramos k-meros que podemos formar dado un alfabeto en particular de tamaño n, entonces tenemos que los posibles k-meros es igual a n^k. Por ejemplo, si consideramos el alfabeto de los 20 aminoácidos estándar y $k = 2$ tenemos que el número total de posibles k-meros es de $20^2 = 400$.

Índice	1	2	3	4	5	6	7	8	9	10	11	12
Secuencia	L	L	G	D	F	F	R	K	S	K	E	K
Distribución positivo, primero							|					
Distribución positivo, 25%							|					
Distribución positivo, 50%								|				
Distribución positivo, 75%										|		
Distribución positivo, 100%												|

Distribución reducida de los aminoácidos (D)

$\mathrm{D}(Positivo, primero, S) = 58.33$

$\mathrm{D}(Positivo, 50\%, S) = 66.67$

$\mathrm{D}(Positivo, 100\%, S) = 100$

$\mathrm{D}(Positivo, 25\%, S) = 58.33$

$\mathrm{D}(Positivo, 75\%, S) = 83.33$

Figura 9. Ejemplo del descriptor molecular distribución reducida de aminoácidos para el grupo positivo del alfabeto Σ_{carga} y la secuencia del péptido antibacteriano (PDB 2FBU).

Dado que el número de *k*-meros es elevado para el alfabeto de los aminoácidos estándar, para el cálculo de descriptores solo consideramos los alfabetos reducidos de hidrofobicidad (Li *et al.*, 2006; Tomii y Kanehisa, 1996) (ver Tabla 3, renglón 5) y los valores de *k* igual a 2 y 3. Por lo tanto, el descriptor denominado dipéptido ($k = 2$) produce nueve valores, mientras que el descriptor tripéptido ($k = 3$) produce 27 valores para el alfabeto reducido hidrofobicidad (ver Tabla 2).

3.1.4. Índice alifático

El índice alifático se define como el volumen relativo ocupado por cadenas laterales de aminoácidos alifáticos de una proteína. Los aminoácidos que pertenecen a este grupo son: alanina (A), valina (V), isoleucina (I) y leucina (L). Los análisis estádisticos del índice alifático, mostrado por Ikai (1980), relevan que este es significativamente mayor en proteínas de bacteria termófilas que en proteínas ordinarias. Por lo tanto, el índice puede considerarse un factor positivo para el aumento de la termoestabilidad de las proteínas globulares (Gasteiger *et al.*, 2005; Ikai, 1980).

El índice alifático (AI) de una proteína puede calcularse de la siguiente forma (Ikai, 1980):

$$AI = x_A + aX_V + b(x_I + x_L), \tag{9}$$

donde x_A, X_V, x_I y x_L son el porcentaje en moles de los residuos A, V, I y L, respectivamente. Como coeficientes a y b se toma el volumen relativo de la cadena lateral de V (*i.e.,* $a = 2.9$) y las cadenas laterales L y V con respecto a la cadena lateral de A (Ikai, 1980). El descriptor AI genera solo un valor para una secuencia en particular.

3.1.5. Carga neta

La carga neta C de un péptido es la suma de los residuos cargados positivamente menos el número de residuos con carga negativa (Klein *et al.*, 1984). Mientras que el promedio de la carga neta MC es el valor de la carga neta divido por la longitud de la secuencia. A continuación, se describen formalmente estos dos conceptos:

Dada una secuencia S de longitud l y una escala de carga sc, la carga neta C y la carga neta promedio MC se definen como sigue (Klein *et al.*, 1984):

$$C(S, sc) = \sum_{i=1}^{l} C_{s_i,sc} \text{ y} \tag{10}$$

$$MC(S, sc) = \frac{1}{l} \sum_{i=1}^{l} C_{s_i,sc} \ , \tag{11}$$

respectivamente, donde $C_{s_i,sc}$ es el valor que toma el residuo s_i en la escala de carga sc. Por ejemplo, si tomamos la escala de carga $sc = KLEP840101$ los valores que tomaría $C_{s_i,sc}$ son los siguientes:

$$C_{s_i,sc} = \begin{cases} -1, & \text{si } x_j \in \{D, E\}, \\ 1, & \text{si } x_j \in \{R, K\}, \\ 0, & \text{otro caso}\,. \end{cases} \tag{12}$$

Las escalas que utilizamos para calcular tanto MC y C fueron recuperadas de la base de datos AAindex (Kawashima y Kanehisa, 2000). Estas fueron tres escalas de carga con los siguientes códigos de acceso: KLEP840101, CHAM830107 y CHAM830108.

Adicionalmente, dado que los AMPs pueden actuar en condiciones distintas de pH,

esto dependiendo de la región donde el objetivo se encuentra, en (Piotto *et al.*, 2012) sugieren calcular la carga de cada péptido a diferentes valores de pH (*i.e.*, pH=5, 7 y 9). La carga Z de un péptido dado un pH en particular se puede calcular de la siguiente manera:

$$Z(pH) = \sum_i N_i \frac{1}{1 + 10^{(pH - pK_{a_i})}} - \sum_j N_j \frac{1}{1 + 10^{(pK_{a_j} - pH)}}, \tag{13}$$

donde, N_i es el número de ocurrencias de los aminoácidos arginina (R), lisina (K) e histidina (H) en la secuencia peptídica S, pK_{a_i} es el valor de pKa para los aminoácidos R, K y H, respectivamente. N_j es el número de veces que los aminoácidos ácido aspártico (D), ácido glutámico (E), cisteína (C) y tirosina (Y) ocurren en la secuencia peptídica x. pK_{a_j} es el valor de pKa para los aminoácidos D, E, C e Y.

3.1.6. Hidrofilicidad, hidrofobicidad e hidropatía

El gran promedio de hidrofilicidad (GRAVY por sus siglas en *Grand Average of Hydropathy*) (Klein *et al.*, 1984), hidrofobicidad (Gasteiger *et al.*, 2005) e hidropatía (Klein *et al.*, 1984) se calcula de forma similar cambiando solo la escala de hidrofilicidad, hidrofobicidad e hidropatía. Esto es, la suma de los valores que toman los residuos dada una escala, dividida entre la longitud del péptido. La fórmula para calcular el gran promedio es como sigue:

$$MH(S, sc) = \sum_{i=1}^{l} C_{s_i, sh}, \tag{14}$$

donde $C_{s_i,sh}$ es el valor que toma el residuo si en la escala sh. Las escalas fueron recuperadas de la base de datos AAindex (Kawashima y Kanehisa, 2000). En la Tabla 4 se presentan las escalas que utilizamos para el cálculo del gran promedio, las cuales se encuentran organizadas por hidrofilicidad, hidropatía e hidrofobicidad, respectivamente.

Tabla 4. Escalas utilizadas para el cálculo del gran promedio de hidrofilicidad, hidropatía e hidrofobicidad. Los códigos que se presentan en la segunda columna fueron recuperados de la base de datos AAIndex (Kawashima y Kanehisa, 2000).

Tipo de escala	**Escala (AAindex)**
Hidrofilicidad	HOPT810101 KUHL950101
Hidropatía	KYTJ820101
Hidrofobicidad	CIDH920102, CIDH920103,CIDH920104, CIDH920105, EISD840101, GOLD730101, JOND750101, MANP780101, PONP800101, PONP800104, PONP800105, PONP800106, PRAM900101, SWER830101, ZIMJ680101, JURD980101, WOLR790101, KIDA850101, CASG920101, ENGD860101, FASG890101, TOSSI2002

3.1.7. Índice de Boman

De acuerdo con (Piotto *et al.*, 2012), el índice de Boman muestra el grado de discriminación entre los péptidos que interactúan con la membrana y los que interactúan con las proteínas. El índice de Boman se define como la suma de las energías libres de las cadenas laterales para la transferencia del ciclohexano al agua y dividida por el número total de residuos del péptido. Un péptido más hidrofóbico tiende a tener un índice negativo, mientras que un péptido más hidrófilo tiende a tener un índice más positivo (Boman, 2003).

3.1.8. Masa molecular y número de aminoácidos

La masa molecular m_w de un péptido S podría estimarse como sigue:

$$w_m = M_N + M_C + \sum_{i=1}^{l} (N_i \times Mi), \tag{15}$$

donde N_i es el número de ocurrencia del aminoácido i en la secuencia s y M_i es el promedio de peso molecular del aminoácido i. Adicionalmente, M_N y M_C se suman al total para tomar en cuenta los terminales: H para el terminal amino y OH para el terminal carboxilo. Para obtener el M_i de los 20 aminoácidos estándar utilizamos la escala con código de acceso GASG760101 de la base de datos AAindex (Kawashima y Kanehisa, 2000).

3.1.9. Índice de inestabilidad

El índice de inestabilidad (II) proporciona una estimación de la estabilidad de una proteína en un tubo de ensayo (Gasteiger *et al.*, 2005; Guruprasad *et al.*, 1990). El II muestra la correlación entre la estabilidad de una proteína y la composición de sus dipéptidos (*i.e.,* 400 posibles dipéptidos). Si el II de una proteína dada es inferior a 40, entonces se considera que es una proteína estable (Gasteiger *et al.*, 2005; Guruprasad *et al.*, 1990). El II se puede calcular como sigue:

$$II = \frac{10}{l} \sum_{i=1}^{l-1} DIWV(x[i]x[i+1]), \tag{16}$$

donde l es la longitud de la secuencia y DIWV(x[i]x[i+1]) es el valor de inestabilidad del dipéptido que comienza en la posición i. Note que DIWV es una matriz de 400×400 que contiene el valor para todos los posibles dipéptidos. Esta matriz puede consultarse en (Guruprasad *et al.*, 1990).

3.1.10. Hidrofobicidad promedio máxima y momento hidrofóbico

La hidrofobicidad promedio máxima se calcula utilizando una ventana de tamaño n en una secuencia de tamaño l como sigue (Eisenberg *et al.*, 1982, 1984):

$$H_{max}(S, sh) = \max_{1 \le i \le l-n+1} MH(s_i, \ldots, s_{i+n}, sh), \tag{17}$$

donde sh es la escala de hidrofobicidad, MH es el gran promedio para la subsecuencia $s_i, \ldots, s_{i+n}$. Por lo tanto H_{max} se define como máximo gran promedio de la secuencia S. La escala de hidrofobicidad que consideramos fue la de EISD840101 (Eisenberg *et al.*, 1984) de la base de datos AAIndex (Kawashima y Kanehisa, 2000). Por lo tanto, para el descriptor H_{max} se produce un valor.

En general, los AMPs adoptan una estructura anfipática (*i.e.,* su estructura tiene regiones hidrofílicas e hidrofóbicas). Una medida cuantitativa de la anfipaticidad de un

péptido es el momento hidrofóbico (HM). El HM se calcula como la suma vectorial de los valores de hidrofobicidad de cada residuo en una secuencia dada. En nuestro caso consideramos el momento hidrofóbico promedio máximo (Eisenberg *et al.*, 1984). A continuación, definimos el HM formalmente.

Dado una secuencia S de longitud l, un ángulo de rotación del péptido θ, una ventana de tamaño n y la escala de hidrofobicidad sh, el momento hidrofóbico promedio máximo (HM_{max}) se calcula como sigue:

$$HM_{max}(S, \theta, n, sh) = \max_{1 \leq i \leq l-n+1} HM(s_i, \ldots, s_{i+n}, \theta, n, sh) \tag{18}$$

y

$$HM(S, \theta, n, sh) = \frac{1}{n}\left(\left[\sum_{i=1}^{n} h_i sin(i \times \theta)\right]^2 + \left[\sum_{i=1}^{n} h_i * cos(i \times \theta)\right]^2\right)^{1/2}, \tag{19}$$

donde h_i es el valor de hidrofobicidad para el residuo i dada la escala sh. Para sh utilizamos la escala normalizada de Eisenberg *et al.* (1982) y para los ángulos de rotación del péptido utilizamos los valores de $\theta = 100, 160, 180$. Por lo tanto, la cantidad de valores que produce el descriptor HM_{max} es de tres.

3.1.11. Punto isoeléctrico

El punto isoeléctrico (pI) se define como el valor de pH para el cual una secuencia dada tiene una carga neta igual a cero (*i.e.,* carga neutra) (Kozlowski, 2016; Gasteiger *et al.*, 2005). Para obtener la carga de un péptido dado utilizamos la Ecuación 13.

3.2. Resultado

Como resultado se implementaron 118 descriptores que producen 268 valores para una secuencia de aminoácidos S. A la biblioteca encargada de generar el conjunto de descriptores la nombramos MODAMP (por sus siglas en inglés de MOlecular Descriptor

for AntiMicrobial Peptides). MODAMP se implementó en Java 8 utilizando el entorno de desarrolló integrado de NetBeans 8.2.

MODAMP recibe como entrada un archivo en formato Fasta, con los péptido representados en secuencias de aminoácidos. Es importante señalar que MODAMP supone que cada secuencia contiene sólo aminoácidos estándar. Cómo salida MODAMP devuelve un archivo CSV, en donde cada renglón contiene los valores de los descriptores para una secuencia en particular, mientras que cada columna representa el valor de un descriptor. Un ejemplo ilustrativo de los archivos de entrada y salida se muestran en el Apéndice A.1.

Capítulo 4. Representación basada en la selección de descriptores moleculares para la clasificación de la actividad antimicrobiana

El modelado cuantitativo de la relación estructura-actividad (QSAR por sus siglas en inglés de *Quantitative Structure–Activity Relationship*) se ha aplicado ampliamente al descubrimiento de AMPs para el desarrollo de modelos cualitativos o cuantitativos que ayudan a determinar la actividad antimicrobiana en los péptidos (Jenssen, 2011). QSAR relaciona matemáticamente las propiedades fisicoquímicas extraídas de los péptidos, denominados descriptores moleculares, con su correspondiente actividad biológica a través de un modelo matemático. Hay dos aspectos cruciales en el modelado QSAR: la selección del conjunto de descriptores que definen la característica de los péptidos de interés y la selección de la técnica de aprendizaje de máquina para crear un modelo (Fernandes *et al.*, 2012; Fjell *et al.*, 2009).

La investigación computacional se ha centrado en el segundo aspecto, en el que se han propuesto varios algoritmos de aprendizaje de máquina (MLA) con este fin. Ejemplos de estos MLAs incluyen análisis de discriminantes (DA) (Thomas *et al.*, 2009), bosques aleatorios (RF) (Thomas *et al.*, 2009; Waghu *et al.*, 2014), máquina de soporte vectorial (SVM) (Thomas *et al.*, 2009; Torrent *et al.*, 2011; Waghu *et al.*, 2014), red neural artificial (ANN) (Fjell *et al.*, 2009; Torrent *et al.*, 2011; Waghu *et al.*, 2014), sistema de inferencia adaptativa neurodifusa (ANFIS) (Fernandez-Escamilla *et al.*, 2004), regresión logística binaria (BLR) (Randou *et al.*, 2013) y los k vecinos más cercanos en su versión difusa (FKNN) (Xiao *et al.*, 2013). En general, los algoritmos propuestos permiten generar modelos con una precisión de predicción de hasta el 96 %. Sin embargo, como se mencionó en el Capítulo 2, existe una discrepancia en las bases de datos utilizadas en estos estudios tanto en tamaño como en las secuencias de péptidos que albergan.

Por el contrario, se ha puesto menos atención a la selección de descriptores moleculares apropiados para representar a los péptidos en la tarea de clasificación de la actividad. Esto, a pesar de que la representación tiene un impacto en el desempeño de los modelos de clasificación. Dado que los descriptores moleculares definen el espacio químico donde se proyecta cada péptido y es en este espacio en donde la clasificación

se lleva acabo. En estudios anteriores, la selección de los descriptores se ha hecho basándose en la intuición química o en las propiedades observadas que dan lugar a la actividad antimicrobiana (Fjell *et al.*, 2012; Torrent *et al.*, 2011). Por otro lado, los estudios recientes emplean procedimientos de selección de características (descriptores) o métodos de filtrado que evalúan independientemente las características de acuerdo a un criterio dado y seleccionan las características principales (Torrent *et al.*, 2011; Waghu *et al.*, 2014; Fernandes *et al.*, 2012). Sin embargo, la mayoría de estos enfoques se centraron en la relación e interacción de los descriptores por pares, mientras que la actividad biológica podría depender de la relación de tres o más de estos.

Por lo tanto, necesitamos un procedimiento de selección de características más exhaustivo para mejorar el rendimiento de los modelos de aprendizaje (Gabere y Noble, 2017). En este trabajo proponemos un método novedoso para seleccionar automáticamente una representación para los péptidos, basada en descriptores moleculares, que realiza eficientemente la clasificación de la actividad antimicrobiana. Para ello, nuestro método combina un algoritmo genético adaptativo basado en especies (SAGA por sus siglas en inglés de *Species Adaptation Genetic Algorithm*) y un modelo de aprendizaje automático para buscar eficazmente soluciones prometedoras y estimar la idoneidad directamente para cada subconjunto de descriptores moleculares. Para evaluar el rendimiento del método propuesto, se utilizaron tres conjuntos de referencia bien conocidos (ver subsecciones 2.3.1.2, 2.3.1.3 y 2.3.1.4). Evaluamos sistemáticamente nuestro método propuesto y lo comparamos con los métodos de predicción de AMPs más avanzados en los tres conjuntos de referencia.

4.1. Planteamiento del problema: selección de características

El objetivo de nuestro enfoque es elegir una representación de péptidos, basado en descriptores moleculares, para discernir entre secuencias AMPs y no AMPs. La elección de los descriptores puede formularse como un problema de selección de subconjuntos de características (FSSP por sus siglas en inglés de *Feature Subset Selection Problem*). En el aprendizaje supervisado, el FSSP puede definirse como: dado un conjunto de datos representado por un conjunto de características, seleccionar aquellas características que son útiles para la tarea de clasificación para un dominio en particular (Guyon y Elisseeff, 2003). En general, la utilidad está dada por la capacidad de predicción del

clasificador, en lugar de la relevancia de las características en el conjunto de datos. A continuación, introducimos la notación sobre la que definimos formalmente el FSSP.

4.1.1. Formulación del FSSP

Considere el conjunto etiquetado $\mathcal{D} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}$ compuesto por n péptidos descritos mediante m descriptores moleculares, en donde cada péptido tiene una etiqueta proveniente del conjunto de etiquetas $Y = \{\text{AMP}, \text{no AMP}\}$. Debe tenerse en cuenta que una instancia, $\mathbf{x}^i$ en $\mathcal{D}$, es un vector m-dimensional en el que la componente x_j^i es el j-ésimo descriptor molecular del péptido i.

Sea un algoritmo de aprendizaje de máquina $\mathcal{I}$, un conjunto de datos etiquetados $\mathcal{D}$ y una función de evaluación J para todos los clasificadores inducidos por $\mathcal{I}$, denotado por $\mathcal{I}(\mathcal{D})$. Entonces, podemos definir el FSSP de la siguiente manera (Kohavi y John, 1997):

$$\begin{aligned} &\underset{X'}{\text{maximizar}} && f(X', \mathcal{D}) = J(\mathcal{I}(\mathcal{D}(X'))) \\ &\text{sujeto a} && X' \subseteq X, \end{aligned} \tag{20}$$

donde $\mathcal{D}(X') \subseteq \mathcal{D}$ es una reducción de $\mathcal{D}$ obtenida tras la eliminación de las características que no están en X' en cada instancia $(\mathbf{x}^\mathbf{i}, y^i) \in \mathcal{D}$. Es importante notar que el subconjunto óptimo de características X_{opt} no es necesariamente único, es decir, es posible lograr el mismo valor en la función de evaluación usando diferentes subconjuntos de características (Kohavi y John, 1997). Sin embargo, inducir un clasificador usando un subconjunto de mayor tamaño es más costoso (*i.e.,* tiempo y recursos computacionales) que usar un subconjunto de menor tamaño. Otra observación es que el tamaño de X_{opt} se desconoce a *priori*, esto hace que FSSP sea más difícil que el problema de seleccionar el subconjunto óptimo cuando el tamaño del mismo es dado como entrada (Webb, 2003).

Desafortunadamente, de manera similar al problema de ponderación de características (ver Capítulo 5), el problema de seleccionar el subconjunto óptimo es un problema NP-difícil (Amaldi y Kann, 1998b).

4.1.2. Enfoque para resolver FSSP

En esta subsección, presentamos un método de envoltura para resolver el FSSP. El método se describe mediante la caracterización de los tres componentes: estrategia de búsqueda, método de estimación del desempeño y un algoritmo de aprendizaje de máquina.

- **Estrategia de búsqueda**. Proponemos un algoritmo genético adaptativo basado en especies para la selección de características, a este algoritmo lo nombramos SAGAFS (por sus siglas en inglés de *Species Adaptation Genetic Algorithm for Feature Selection*) . SAGAFS es una versión adaptada de dos algoritmos bien conocidos: un algoritmo genético (GA) y un algoritmo evolutivo de representación de longitud variable (VLREA) (Zebulum *et al.*, 2000). GA es comúnmente recomendado para problemas de selección de características a gran escala (*i.e.*, 50 o más características candidatas) (Huang *et al.*, 2007). Por otro lado, VLREA es adecuada para problemas en los que la longitud de la solución contribuye a su aptitud, como ocurre en nuestro caso. Hasta donde sabemos, esta es la primera vez que se aplica un algoritmo evolutivo VLR al problema de selección de características. El algoritmo SAGAFS propuesto incluye una representación de longitud variable y una estrategia de espacios vecinos para muestrear eficientemente el vasto espacio de búsqueda.
- **Método de estimación del desempeño**. Utilizamos la validación cruzada de k pliegues para estimar, en promedio, el coeficiente de correlación de Matthew (MCC) del clasificador inducido por un algoritmo de aprendizaje de máquina y un conjunto de datos.
- **Algoritmo de aprendizaje de máquina**. Para la generación de un clasificador binario se utilizaron dos algoritmos de aprendizaje automático: el primero, un clasificador lineal, este es una máquina de soporte vectorial (SVM-L); el segundo, un clasificador no lineal, bosques aleatorios (RF).

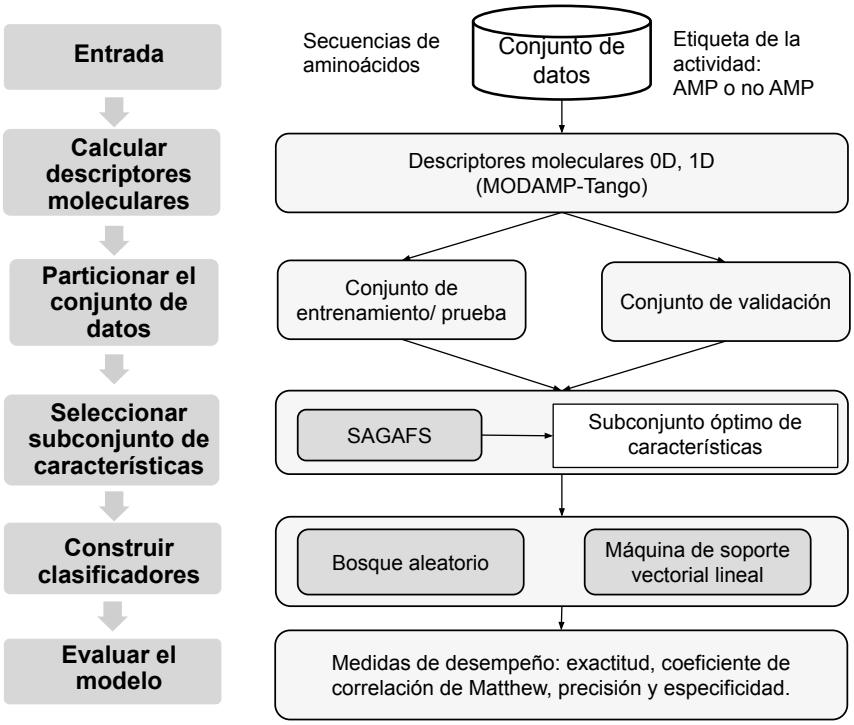


Figura 10. Esquema general para la selección automática de descriptores moleculares para la clasificación eficiente de la actividad antimicrobiana.

4.2. Materiales y Métodos

La metodología adoptada en este trabajo se describe en los siguientes apartados y en la Figura 10 se muestra el esquema.

4.2.1. Conjunto de datos

Se consideraron tres conjuntos de datos de referencia ampliamente utilizados en la tarea de clasificación binaria de la actividad antimicrobiana en péptidos (Xiao *et al.*, 2013; Fernandes *et al.*, 2012; Thomas *et al.*, 2009) (para más detalles de cómo están formados los conjuntos ver Capítulo 2.3.1). Se utilizaron estos conjuntos de datos para medir, de forma imparcial, el rendimiento de los descriptores moleculares obtenidos por SAGAFS. Los conjuntos de datos son: DAT1 propuesto por Fernandes *et al.* (2012), DAT2 propuesto por Thomas *et al.* (2009) y DAT3 propuesto por Xiao *et al.* (2013). Dado que en estos conjuntos existen secuencias cortas (*i.e.,* de longitud menor a 10 aa) decidimos eliminar las secuencias que tiene un tamaño fuera del intervalo de 10 a 100 aminoácidos. La Figura 11 muestra la superposición entre estos conjuntos de datos; puede observarse que el traslape se da únicamente entre el conjunto de péptidos an-

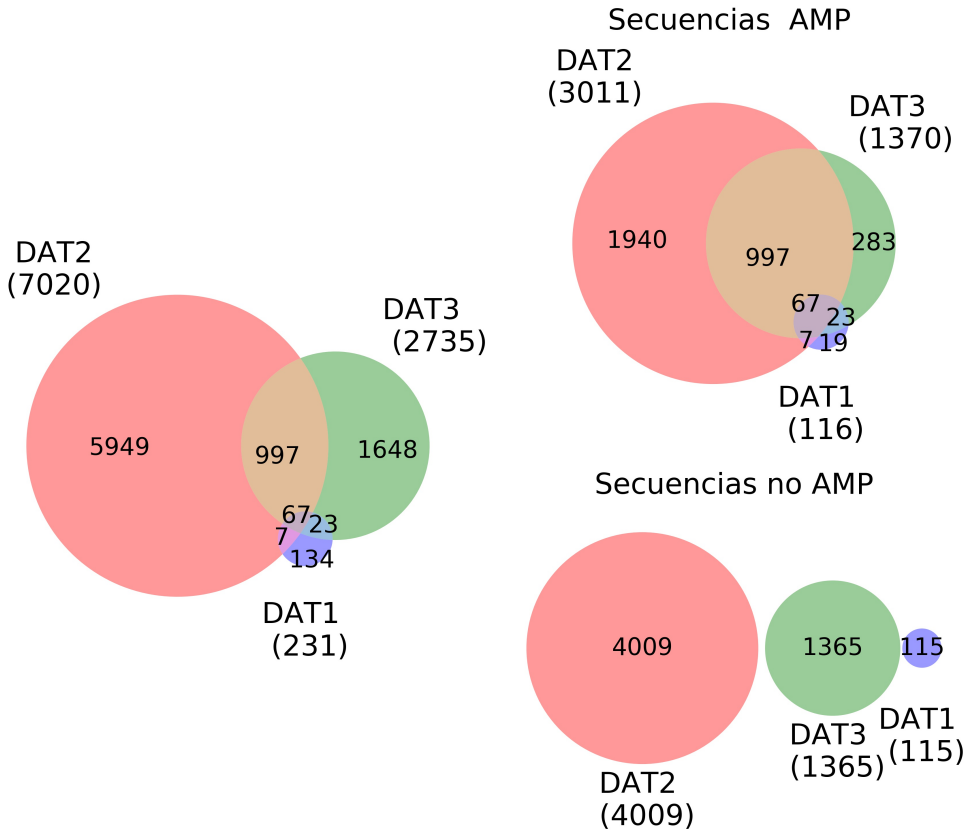


Figura 11. Diagrama de Venn de los conjuntos de datos de referencia considerados para la prueba de SAGAFS. El nivel de superposición entre el conjunto de datos DAT1 (Fernandes *et al.*, 2012), DAT2 (Thomas *et al.*, 2009) y DAT3 (Xiao *et al.*, 2013) correspondiente sólo a AMPs, *i.e.*, entre estos tres conjuntos de datos, no hay intersección con péptidos no antimicrobianos.

timicrobianos, aunque los tres conjuntos de datos utilizaron una metodología similar para recuperar secuencias no antimicrobianas.

4.2.2. Cálculo de descriptores moleculares

Para calcular los descriptores moleculares, utilizamos dos paquetes de software diferentes: Tango (Rousseau *et al.*, 2006; Fernandez-Escamilla *et al.*, 2004; Linding *et al.*, 2004b) y la herramienta interna MODAMP (por sus siglas en inglés de *MOlecular Descriptor for AntiMicrobial Peptides*) (ver Capítulo 3). Tango se utilizó para calcular los descriptores relacionados con la estructura secundaria, mientras que el MODAMP (Beltrán *et al.*, 2017) se utilizó para calcular los descriptores restantes. En total se calcularon cuatro descriptores moleculares usando Tango (Rousseau *et al.*, 2006; Fernandez-Escamilla *et al.*, 2004; Linding *et al.*, 2004b) y otros 268 con la herramienta MODAMP (Beltrán *et al.*, 2017).

En este paso, suponemos que cada péptido, del conjunto de datos de entrada, es una secuencia válida $S_i = s_1, \ldots, s_l$, es decir, cada símbolo s_j proviene del alfabeto de los 20 aminoácidos estándar. Considerando el conjunto de descriptores moleculares $\mathcal{X} = \{X_1, \ldots, X_m\}$ para $m = 272$, convertimos cada secuencia S_i en un vector $\mathbf{x}^i = \left[x_1^i, x_2^i, \ldots, x_m^i\right]^T$, cada componente x_j^i codifica el valor para el descriptor molecular X_j de la secuencia S_i.

4.2.3. Algoritmo de selección de subconjuntos de características

4.2.3.1. Representación de la solución

Diseñar una representación adecuada para las soluciones candidatas (*i.e.,* subconjunto de características) es un paso esencial en el algoritmo genético (GA) (Eiben *et al.*, 2003). La representación en un GA define una asignación del espacio de las soluciones candidatas, denominado espacio fenotípico, al espacio genotípico, en donde se lleva a cabo la búsqueda de la solución óptima. Para el problema FSSP (ver Subsección 4.1.1), el espacio fenotípico está dado por los posibles subconjuntos que se pueden formar del conjunto de características excluyendo al conjunto vacío. Por lo general, cada subconjunto en el espacio genotípico se representa como una cadena binaria de longitud fija (Kabir *et al.*, 2011; Huang *et al.*, 2007), en donde cada posición toma el valor de uno si la característica es parte de la solución y cero en otro caso. La principal desventaja de esta representación es que genera cromosomas extensos, en donde sólo unos pocos bits codifican las características de una solución candidata. Por lo anterior, consideramos una representación de longitud variable (VLR por sus siglas en inglés de *Variable Length Representation*) que permite codificar sólo las características relacionadas con la solución candidata y en consecuencia, podemos obtener una representación más compacta.

Representación. Un cromosoma g es un subconjunto de números enteros que codifica el índice de cada característica seleccionada. Entonces un genotipo dado, $g = \{g_1, g_2, \ldots, g_k\}$, donde $g_i \in \{1, \ldots, m\}$ y $k \leq m$, representa el subconjunto $X_g = \{X_{g1}, X_{g2}, X_{g3}, \ldots, X_{gk}\}$. A continuación, mostramos un ejemplo de un individuo y su solución correspondiente (fenotipo).

$$\begin{array}{lll} \text{Cromosoma} & & \text{Solución candidata} \\ g = \{1, 3, 5\} & \rightarrow & X_g = \{X_1, X_3, X_5\} \end{array}$$

4.2.3.2. Función de aptitud

La calidad de un subconjunto X_g se mide de forma indirecta a través del desempeño de un modelo de clasificación. El clasificador es inducido por un algoritmo de aprendizaje de máquina, el cual es entrenado utilizando el conjunto de datos con sólo las características codificadas en X_g.

Función de aptitud. Dado un subconjunto X_g representado por un cromosoma g, la aptitud de g se define como,

$$f(X_g) = J(\mathcal{D}(X_g)) + \lambda \frac{|X_g|}{m}, \tag{21}$$

donde,

$$J(\mathcal{D}(X_g)) = \frac{1}{k} \sum_{i=1}^{k} |MCC_i(\mathcal{I}(\mathcal{D}(X_g) - \mathcal{D}_i(X_g), \mathcal{D}_i(X_g)))|.$$

Aquí, $J(\mathcal{D}(X_g))$ es el coeficiente de correlación de Matthew (MCC) estimado por la validación cruzada de k pliegues sobre el conjunto de datos $\mathcal{D}(X_g)$. $\mathcal{D}(X_g)$ es la reducción del conjunto de entrada a nivel de características, en donde se eliminan de $\mathcal{D}$ las características que no se encuentran en X_g. Además, $|.|$ es el valor absoluto de MCC_i obtenido por el clasificador inducido $\mathcal{I}$ para del conjunto de validación $\mathcal{D}_i(X_g)$. Es importante notar que el conjunto de datos $\mathcal{D}(X_g)$ se divide en k partes mutuamente excluyentes de igual tamaño, en donde se utilizan $k-1$ particiones para entrenar $\mathcal{I}$, es decir, $\mathcal{D}(X_g) - \mathcal{D}_i(X_g)$. El segundo término en la Ecuación 21 es un criterio de desempate que beneficia a las soluciones más simples, esto es, subconjuntos de menor cardinalidad. Aquí, λ es un valor en el intervalo $[10^{-2}, 10^{-4}]$ y m es la cardinalidad del universo de características.

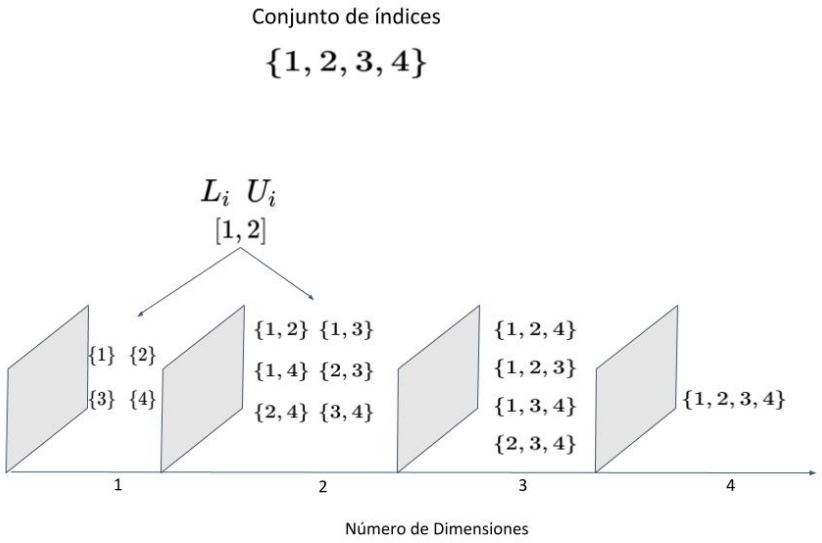


Figura 12. Ejemplo del espacio genotípico para un conjunto de cuatro características. La imagen muestra como el espacio se divide en subespacios de acuerdo con el tamaño de los subconjuntos. *i.e.,* conjuntos con cardinalidad uno, dos, tres y cuatro, respectivamente. Además, se muestran las cotas (L_i y U_i) para la inicialización de los individuos.

4.2.3.3. Principales pasos de SAGAFS

Inicialización de la población. La población inicial de N_{pop} individuos, $P(t)$ con $t = 1$, es generada al azar. Para cada cromosoma, $g \in P(t)$, se seleccionan al azar k valores enteros de los m disponibles. Es importante señalar que k está restringido por la cota inferior L_i y la cota superior U_i. Estos límites se emplean para restringir el tamaño de los individuos iniciales, lo que nos permite delimitar los subespacios a visitar inicialmente. Por ejemplo, en la Figura 12 se muestra el espacio genotípico para un conjunto de entrada de cuatro características, en donde sólo para generar la población inicial se pueden tomar individuos de los subespacios en donde se encuentran individuos de cardinalidad uno y dos, respectivamente.

Posterior a la generación de los individuos, el valor de aptitud de cada g en $P(t)$ es calculado de acuerdo con la función de aptitud descrita en la Ecuación 21.

Selección de los padres y recombinación. De la población actual $P(t)$, se seleccionan μ individuos utilizando el torneo binario con reemplazo (Eiben *et al.*, 2003). Los individuos obtenidos son añadidos al conjunto de padres denotado como $M(t)$.

Después, reordenamos $M(t)$ de forma aleatoria.

Por cada par consecutivo de padres, *i.e.*, g_i y g_{i+1} en $M(t)$ para $i \leq \mu - 1$ (donde μ es el número de padres), se generan los hijos o_i y o_{i+1}, y se añaden al conjunto $O(t)$. Para generarlos, primero se produce un número aleatorio r_i distribuido uniformemente en el intervalo (0,1). Enseguida, dado r_i y la probabilidad de cruzamiento p_c, se aplica una de las siguientes dos operaciones:

- Si $r_i \leq p_c$ entonces se aplica la recombinación entre los padres g_i y g_{i+1} usando el operador de cruzamiento *subset size-oriented common feature crossover operator* (SSOCF) (Emmanouilidis *et al.*, 2000).
- En otro caso, se copia el cromosoma del padre g_i al hijo o_i y el del padre g_{i+1} al hijo o_{i+1}.

El operador de cruzamiento SSOCF originalmente fue propuesto para representaciones de longitud fija (Emmanouilidis *et al.*, 2000). Sin embargo, en SAGAFS lo hemos adaptado para individuos de longitud variable. El SSOCF tiene la ventaja de preservar las características comunes entre los padres en su descendencia.

Mutación. Para cada hijo en el conjunto de hijos, denotado como $O(t)$, se aplica el operador de mutación k-indel con una probabilidad p_m. Convencionalmente, p_m es un valor definido por el usuario y es estático, esto es p_m no cambia durante la ejecución del algoritmo genético. Sin embargo, en SAGAFS p_m se estima dinámicamente mediante la mutación adaptable propuesta por (Smullen *et al.*, 2014). Este método se utiliza para aumentar p_m cuando la población actual $P(t)$ está por encima de un umbral de similitud (*i.e.*, la población tiene baja diversidad), de lo contrario p_m se reduce. En particular, el valor de similitud de una población está dado por $s(P(t)) = \frac{s}{N_{pop}}$, donde s es el número de individuos idénticos en $P(t)$. La probabilidad de mutación adaptable p_m se calcula de la siguiente manera:

$$p_m = p_0 + sgn(s) \times \sigma, \tag{22}$$

donde

$$sgn(s) = \begin{cases} -1 & \text{si } s < \theta \\ 0 & \text{si } s = \theta \\ 1 & \text{si } s > \theta \end{cases}$$

aquí, θ es el umbral de similitud y es un hiperparámetro del algoritmo SAGAFS. Por otro lado, p_0, es la probabilidad inicial de mutación y σ es el tamaño de cada paso.

Por otra parte, diseñamos el operador de mutación k-inserciones/deleciones (k-indels) con el objetivo de variar el tamaño de un hijo en particular. El operador de mutación k-indels selecciona aleatoriamente un entero k en el intervalo de $[1, m]$. Después, cada número entero seleccionado se inserta o se elimina del hijo, esto dependiendo si el número entero está o no en el cromosoma. Para ilustrar este operador, presentamos el siguiente ejemplo:

$$o = \{2, 3, 8, 10\} \quad \rightarrow \quad o = \{3, \mathbf{6}, 8, \mathbf{9}, 10\} \quad \text{donde } k \in \{2, 6, 9\}.$$

En este ejemplo la característica número 2 se borra mientras que las características número 6 y 9 se añaden.

Después que los hijos en $O(t)$ han sido mutados con una probabilidad p_m, la función de aptitud descrita en la Ecuación 21 es calculada a cada hijo.

Selección de los sobrevivientes. Se aplicó el método de selección por elitismo (Eiben *et al.*, 2003) para seleccionar a los individuos que pasarán a la siguiente generación (*i.e.,* $t + 1$). Este método consiste en primero unir los N_{pop} individuos en $P(t)$ y μ individuos en $O(t)$, después se ordenan de manera descendente con respecto a sus valores de aptitud. Por último, se selecciona a los N_{pop} individuos como aquellos que pasarán a formar parte de la siguiente generación $P(t + 1)$.

Condición de parada de SAGAFS. El algoritmo SAGAFS puede detenerse de acuerdo con dos condiciones: la primera, es que el número de generaciones t sea mayor al número máximo de generaciones n_g; la segunda, es que la mejor aptitud se repita de manera consecutiva el mismo número de generaciones sin mejora n_{gwi}. La segunda condición de paro tiene la ventaja de detener oportunamente la ejecu-

ción de SAGAFS cuando el algoritmo se encuentra estancado o la solución óptima es encontrada.

En resumen, el algoritmo SAGAFS entrega como salida tanto el mejor subconjunto de características X_{opt} y su valor de aptitud $f(X_{opt})$.

4.2.4. Algoritmos de clasificación

Una vez que se obtiene el subconjunto óptimo X_{opt}, el siguiente paso es reducir la matriz de descriptores $\mathcal{D}$ a $\mathcal{D}(X_{opt})$. Como se describió en la Subsección 4.1.1, $\mathcal{D}(X_{opt})$ es la reducción obtenida tras la eliminación de las características que no están en X_{opt} en cada instancia de x_i en $\mathcal{D}$. Después, con el objetivo de medir la calidad de las soluciones encontradas y comparar los resultados con otros clasificadores existentes de AMPs (ver Subsección 4.3.2), construimos los modelos de clasificación binaria de la actividad antimicrobiana . Para la construcción de los modelos utilizamos los mismos algoritmos de aprendizaje de máquina que se utilizaron en el método de envoltura, esto son los bosques aleatorios (RF por sus siglas en inglés de *Random Forest*) y la máquina de soporte vectorial lineal (SVM-L por sus siglas en inglés de *Support Vector Machine-Linear*).

4.2.5. Detalles de implementación

Todos los experimentos se realizaron bajo las siguientes condiciones: sistema operativo Ubuntu 16.04 LTS; CPU: Intel i7 a 2.40GHz; y memoria RAM: 12 GB.

El algoritmo SAGAFS se implementó en Java 8 utilizando el entorno de desarrollo integrado NetBeans 8.2. Los hiperparámetros principales para SAGAFS se muestran en la Tabla 5. Es importante notar que en la Tabla 5, sólo para algunos hiperparámetros (*e.g.,* p_0, N_{pop}) los valores son diferentes para el conjunto de datos DAT1, mientras que para el resto de los hiperparámetros los valores aplican para los tres conjuntos de datos (*i.e.,* DAT1, DAT2, DAT3).

Los algoritmos de aprendizaje de máquina RF y SVM-L se implementaron usando la librería para Java Weka 3.8. Weka 3.8 es una colección de algoritmos de aprendizaje

Tabla 5. Valores de parámetros usados en SAGAFS

Símbolo	Valores	Descripción
Parámetros de control en el cruzamiento SSOCF y la mutación adaptable *k*-indels		
p_c	0.7	Probabilidad de cruzamiento
p_0	0.05, 0.3 [a]	Probabilidad de mutación inicial
σ	0.01	Paso de la mutación
θ	0.15	Umbral de similitud
Tiempo de ejecución y condición de paro		
N_{pop}	48, 32 [a]	Tamaño de la población
N_{gen}	500	Máximo número de generaciones
N_{gwi}	0.1*N_{gen}	Número de generaciones sin mejora
N_r	30	Número de ejecuciones del algoritmo
Parámetros de control en SAGAFS		
$\|X\|$	272	Número de características de entrada
L_i	0.01*$\|X\|$	Cota inferior
U_i	0.05*$\|X\|$, 0.11[a]*$\|X\|$	Cota superior

[a] Valor para el conjunto de datos DAT1

de máquina que contiene herramientas para la preparación de datos, clasificación y visualización. Los hiperparámetros que se utilizaron para cada uno de los algoritmos de aprendizaje de máquina RF y SVM-L se encuentran descritos en el Apéndice A.2.

4.3. Resultados

Para evaluar la eficacia de nuestro enfoque, llamado SAGAFS, realizamos experimentos con tres conjuntos de datos. Anteriormente, estos datos se han usado para evaluar los modelos de clasificación de AMPs propuestos en el estado del arte (Fernandez-Escamilla *et al.*, 2004; Thomas *et al.*, 2009; Waghu *et al.*, 2014; Xiao *et al.*, 2013). Primero, ejecutamos 30 veces el algoritmo SAGAFS para cada conjunto de datos. Después, seleccionamos la mejor solución encontrada por cada conjunto de datos y la comparamos con los resultados de los métodos del estado del arte en clasificación de AMPs. La comparación de nuestros resultados con los métodos del estado del arte se realizó utilizando el mismo conjunto de datos. A continuación, se describe cuál conjunto de datos se utilizó para cada comparación: DAT1 se usó para comparar el método ANFIS (Fernandes *et al.*, 2012); DAT2 para el método de CAMP (Waghu *et al.*,

Tabla 6. Desempeño promedio de las mejores soluciones obtenidas por SAGAFS para los tres conjuntos de datos de referencia después de 30 ejecuciones. Para cada métrica se muestra el valor promedio y la desviación estándar entre paréntesis.

Conjunto de datos	MLA *	Acc(%)	Sens	Spec	F-score	MCC	AUC
DAT1	SVM-L	92.70(±1.51)	0.91(±0.05)	0.94(±0.04)	0.93(±0.02)	0.86(±0.03)	0.93(±0.02)
	RF	93.76(±1.01)	0.93(±0.02)	0.94(±0.02)	0.94(±0.01)	0.88(±0.02)	0.95(±0.01)
Dat2	SVM-L	82.01(±0.73)	0.81(±0.03)	0.83(±0.03)	0.82(±0.01)	0.63(±0.02)	0.82(±0.01)
	RF	92.50(±0.40)	0.91(±0.04)	0.93(±0.03)	0.92(±0.00)	0.85(±0.01)	0.97(±0.00)
Dat3	SVM-L	95.12(±0.42)	0.94(±0.01)	0.96(±0.00)	0.95(±0.00)	0.90(±0.01)	0.95(±0.00)
	RF	96.28(±0.61)	0.96(±0.01)	0.96(±0.01)	0.96(±0.01)	0.93(±0.01)	0.99(±0.00)

* Algoritmo de aprendizaje de máquina (MLA): RF=*Random Forest*; SVM-L=*Support Vector Machine-Linear*.

2014); DAT3 para comparar los métodos iAMP-2l (Waghu *et al.*, 2014) y MLAMP (Lin y Xu, 2016).

4.3.1. Selección del modelo

A continuación, presentamos el comportamiento promedio de las mejores soluciones encontradas por SAGAFS, esto después de ejecutarlo 30 veces. Por lo tanto, para comparar las mejores soluciones encontradas, se emplearon las siguientes métricas: sensibilidad (Sens), especificidad (Spec), *F-score*, coeficiente de correlación de Matthew (MCC) y área bajo la curva ROC (AUC por sus siglas en inglés).

El comportamiento promedio de SAGAFS, después de 30 ejecuciones para cada conjunto de datos, se muestra en la Tabla 6. En general, los resultados muestran un desempeño similar en las 30 ejecuciones (*i.e.*, desviación estándar pequeña). Los resultados muestran que, en promedio, los modelos de clasificación generados por RF tienen mejor desempeño para todas las métricas. Además, el mejor desempeño obtenido por conjunto de datos fue usando DAT3, esto con Acc(%) de 96.28 ± 0.61 y el MCC fue 0.93 ± 0.01. Adicionalmente, para determinar el efecto de SAGAFS en la eficiencia de los modelos de clasificación, comparamos el rendimiento de tipos de clasificadores generados por el mismo algoritmo de aprendizaje de máquina, unos aplicando SAGAFS y los otros utilizando todas las características de entrada (*i.e.,* línea base). En esta dirección, los resultados muestran que, en promedio, las mejores soluciones encontradas por SAGAFS tienen mejores valores en MCC que los modelos producidos utilizando todas las características (ver Figura 13).

Por otra parte, el porcentaje de reducción del número de descriptores moleculares se muestra en la Figura 13. Los resultados muestran que la reducción en el número de descriptores moleculares es, en promedio, de al menos 80 %, es decir, los mejores

subconjuntos de características encontrados por SAGAFS tienen a lo más 54 características de un total de 272. En particular, las soluciones que tienen mejor MCC, presentan una reducción promedio del 90 % en el número de características.

Los índices de las características más frecuentemente seleccionadas entre las 30 mejores soluciones por conjunto de datos se muestran en la Figura 14. La lista completa de los nombres de las características de cada índice se muestra en la información complementaria (archivo con nombre MODAMP_ListaDeDescriptores.xls). Por ejemplo, para DAT1, la característica más frecuentemente seleccionada fue la agregación *in vitro* (*i.e.,* característica con índice 268) (ver primera columna en Figura 14). Para DAT2, las características más seleccionadas fueron KLEP840101_CH y la frecuencia de aminoácidos de cisteína en la secuencia de péptidos (*i.e.,* índice 1), respectivamente. Por último para DAT3, las características más frecuentes fueron la frecuencia de metioninas y de cisteínas (índices 10 y 1).

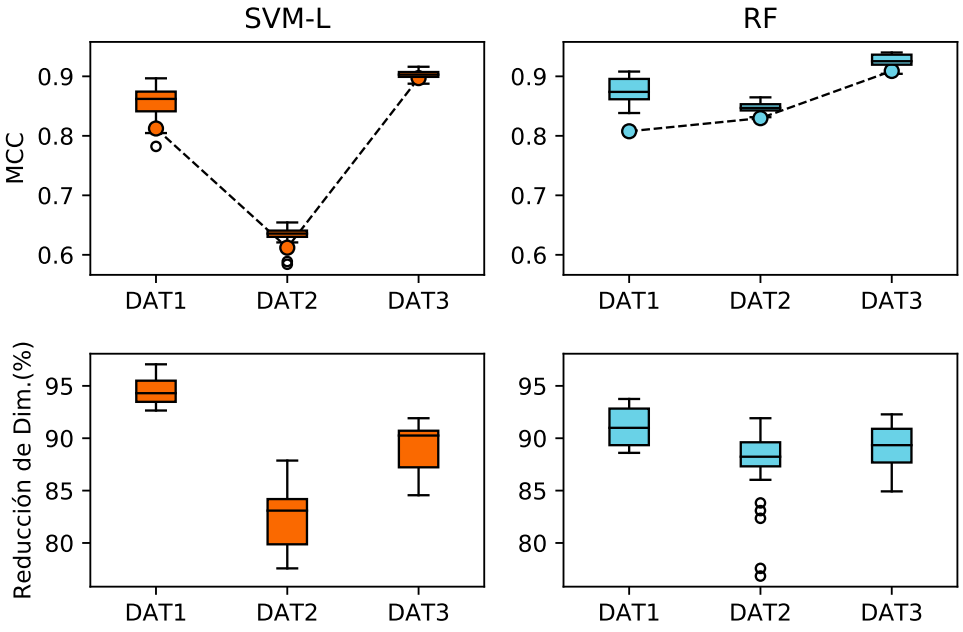


Figura 13. Comparación de desempeño entre las mejores soluciones obtenidas por SAGAFS+SVM-L y SAGAFS+RF después de 30 ejecuciones. La línea punteada indica el MCC para el modelo de línea base.

4.3.2. Comparación con los clasificadores AMP existentes

El mejor modelo generado por nuestro enfoque SAGAFS, para cada conjunto de datos, fue comparado con otros clasificadores existentes de AMPs. Esta comparación

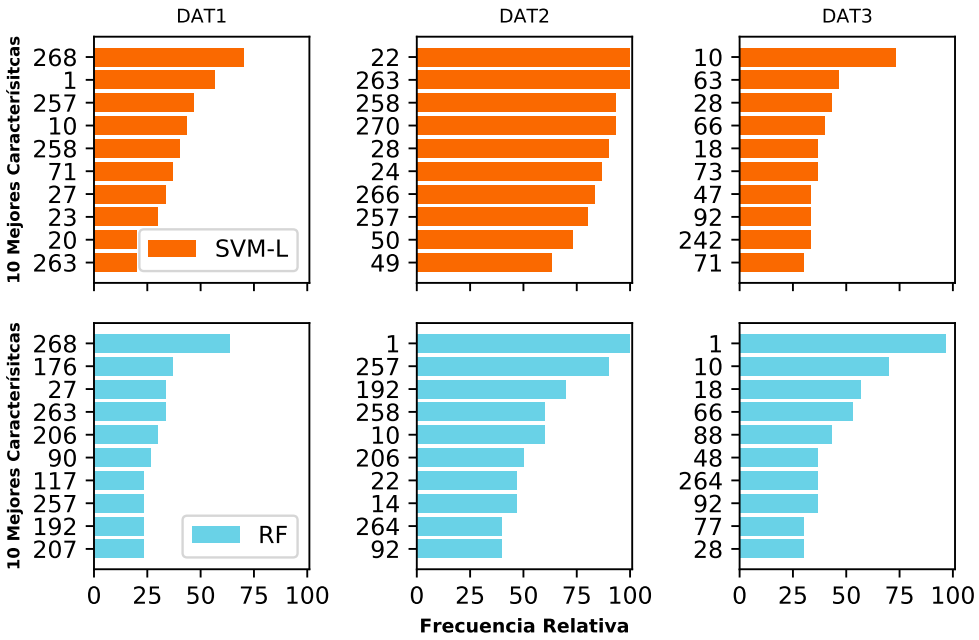


Figura 14. Características más frecuentemente seleccionadas para SAGAFS en cada conjunto de datos. Las gráficas superiores representan los índices de la característica más frecuente para el modelo generado por bosques aleatorios (RF), mientras que las gráficas inferiores muestran los índices para los modelos generados por las máquinas de soporte vectorial lineal (SVM-L).

se realizó utilizando el mismo método de estimación del desempeño y el conjunto de datos. Lo anterior es con el fin de hacer una comparación justa con los métodos del estado del arte. Por ejemplo, si el método propuesto por (Waghu *et al.*, 2014) utiliza el conjunto de datos DAT2, comparamos este método con el mejor obtenido por SAGAFS para este conjunto utilizando el mismo método de estimación del desempeño, *i.e.,* si al conjunto de datos DAT2 lo dividen 80 % para entrenamiento y el resto para pruebas, nosotros lo utilizamos en esta forma. En este sentido es importante señalar que para el ejemplo anterior, la partición de pruebas no se utiliza para encontrar al mejor individuo en el algoritmo SAGAFS.

El mejor modelo para el conjunto DAT1 se comparó con el mejor modelo presentado por Fernandes *et al.* (2012), este utiliza un sistema de inferencia adaptativa neurodifusa (ANFIS por sus siglas en inglés de *Adaptive Neuro-Fuzzy Inference System*) para clasificar péptidos antimicrobianos de una forma binaria. En la Tabla 7, se muestran los resultados entre los modelos de SAGAFS y los de Fernandes *et al.* (2012). Los resultados se muestran en cuatro partes, tres de estas partes corresponden a las particiones que se hicieron a DAT1, esto es 50 % para el entrenamiento, 25 % prueba, 25 % validación. Para el algoritmo SAGAFS solo se utilizó las particiones de entrenamiento y

Tabla 7. Comparación de desempeño entre nuestra propuesta SAGAFS y ANFIS (Fernandes *et al.,* 2012) para el conjunto de datos DAT1.

Método	MLA[a]	Conjunto de datos	Acc(%)	Sens	Spec	F1-score	MCC
Fernandes *et al.* (2012)	ANFIS	Entrenamiento	96.23	1.00	0.93	0.96	0.93
		Prueba	100	1.00	1.00	1.00	1.00
		Validación	94.34	0.96	0.92	0.95	0.89
		General	96.73	**0.99**	0.95	**0.97**[b]	**0.94**
SAGAFS	RF	Entrenamiento	100.0	1.00	1.00	1.00	1.00
		Prueba	84.48	0.88	0.79	0.84	0.70
		Validación	100	1.00	1.00	1.00	1.00
		General	**96.89**	0.97	**0.97**	**0.97**	**0.94**

[a] Algoritmo de aprendizaje de máquina (MLA): bosques aleatorios (RF); sistema de inferencia adaptativa neurodifusa (ANFIS).
[b] La letra en negrita indica el mejor valor por medida.

validación. En general, para las métricas de exactitud (ACC(%)) y especificidad (Spec), nuestro modelo presenta un mejor desempeño que ANFIS (Fernandes *et al.*, 2012), mientras que para las métricas de sensibilidad (Sens), F1-score y MCC muestra un desempeño comparable con el modelo de Fernandes *et al.* (2012). Otra comparación que se realizó fue entre nuestro mejor modelo y los modelos CAMP presentados por Waghu *et al.* (2014). En esta comparación se utilizó el conjunto de datos DAT2, este se dividió en dos partes: la primera parte tiene al 70 % de las instancias de DAT2 para entrenamiento, esta parte fue la que se utilizó en el algoritmo SAGAFS con una validación cruzada de 10 pliegues; la segunda parte es para pruebas y tiene un tamaño del 30 % de DAT2. En la Tabla 8 se muestra la comparación entre SAGAFS y CAMP (Waghu *et al.*, 2014), en donde en las métricas de MCC y ACC(%) ambos métodos muestran un desempeño similar. De manera particular, el método de CAMP que utiliza RF supera en la métrica de sensibilidad a nuestro modelo, mientras que nuestro modelo es más específico que el de CAMP. En la Tabla 9 se compara nuestro modelo con iAMP-2L (Xiao *et al.*, 2013) y MLAMP (Lin y Xu, 2016) para el conjunto de datos DAT3. El desempeño alcanzado por nuestro modelo (*i.e.,* SAGAFS) supera ampliamente en todas las métricas a los métodos iAMP-2L y MLAMP. En particular, nuestro modelo generado con el conjunto de datos DAT3 es el que mejor exactitud tiene.

4.3.3. Discusión

En resumen, se ha propuesto un algoritmo evolutivo novedoso y eficaz para el problema de selección de subconjuntos de características para la clasificación de péptidos antimicrobianos. El enfoque combina dos algoritmos, un algoritmo genético y un algoritmo evolutivo de longitud variable, y además utiliza una función objetivo para

Tabla 8. Comparación de desempeño entre nuestra propuesta SAGAFS y CAMP (Waghu *et al.,* 2014) en el conjunto de datos DAT2.

Método	MLA[a]	MCC	Prueba (%)				Acc(%)
		Entrenamiento	Prueba	Sens	Spec	Acc	
Waghu *et al.* (2014)	RF	0.82	**0.84**	**90.8**	93.7	92.5	**93.4**
	SVM	**0.91**[b]	0.83	89.7	93.1	91.6	92.6
	ANN	0.72	0.72	82.9	88.9	86.3	86.9
SAGAFS	RF	0.87	**0.84**	88.5	**95.14**	92.4	93.3

[a] Algoritmo de aprendizaje de máquina (MLA): bosques aleatorios (RF); máquina de soporte vectorial con kernel polinomial grado 4 (SVM); redes neuronales artificiales (ANN).
[b] La letra en negrita indica el mejor valor por medida.

Tabla 9. Comparación de desempeño entre nuestro método SAGAFS, iAMP-2L (Xiao *et al.,* 2013) y MLAMP (Lin y Xu, 2016) en el conjunto DAT3.

Método	MLA[a]	Sens(%)	Spec(%)	Acc(%)	MCC[a]
iAMP-2L (Xiao *et al.*, 2013)	FKNN	87.13	86.03	86.32	0.727
MLAMP (Lin y Xu, 2016)	RF	77.0	94.60	89.90	0.737
SAGAFS	RF	**96.64**[b]	**97.36**	**97.00**	**0.940**

[a] Algoritmo de aprendizaje de máquina (MLA): bosques aleatorios (RF); *k*-vecinos más cercanos difuso (FKNN).
[b] La letra en negrita indica el mejor valor por medida.

evaluar la calidad de las características seleccionadas. Esta función combina el valor de MCC del clasificador con el número de descriptores seleccionados. Los resultados de los experimentos computacionales sugieren que el método propuesto, SAGAFS, es capaz de encontrar una representación de los péptidos capaz de generar modelos que superan a los métodos de última generación que están disponibles públicamente para la predicción de AMPs.

Capítulo 5. Representación basada en la ponderación de descriptores moleculares para la clasificación de la actividad antimicrobiana

En este capítulo se aborda el problema de seleccionar automáticamente la representación adecuada de los péptidos, basados en descriptores moleculares, para la tarea de determinar si un péptido es antimicrobiano o no. Para este propósito, proponemos una adaptación al exitoso método para la selección de características propuesto por Paul y Das (2015). Este método utiliza un enfoque de filtrado basado en la ponderación de las características. En general, la idea es asignar pesos a estas, de tal forma que instancias de diferentes clases tiendan a estar separadas una de otras, mientras que las instancias de la misma clase tiendan a estar cercanas entre sí. Un problema de esta idea es que si escalamos los descriptores con el fin de minimizar la distancia entre instancias de la misma clase, la distancia entre instancias con diferente clase también se minimizarán (*i.e.,* la distancia entre clase se verá afectada) y viceversa. Estos hechos indican que existe un compromiso entre ambas distancias, por lo que al mejorar una distancia se empeora la otra. Por lo tanto, el problema de ponderación de características es formulado como un problema de optimización multiobjetivo, para el cual no existe una solución que produzca un óptimo para ambas distancias sino un conjunto de soluciones óptimas (*i.e,* un conjunto de vectores de pesos).

En este trabajo, el enfoque propuesto por Paul y Das (2015) ha sido adaptado con el objetivo de ajustarlo a las particularidades del conjunto de datos utilizado para la clasificación binaria de los péptidos antimicrobianos. En nuestra propuesta, tomamos en cuenta que moléculas con estructura similar tienden a poseer una actividad biológica similar (Cai *et al.*, 2013). Por esta razón, consideramos innecesario minimizar la distancia entre péptidos no antimicrobianos debido a que pueden tener diferentes actividades biológicas. Lo anterior, consecuencia de cómo son recuperados los conjuntos de datos negativos (*i.e.*, no AMPs).

Con el objetivo de comparar nuestra formulación con respecto la formulación original (Paul y Das, 2015), evaluamos las mejores soluciones obtenidas utilizando el conjunto de datos de Fernandes *et al.* (2012) para la clasificación de AMPs (*i.e.,* DAT1). La prueba de concepto de nuestra formulación mostró una buena capacidad para la cla-

sificación binaria de AMPs (Beltrán *et al.*, 2017). En el presente trabajo ampliamos los resultados con la evaluación de seis conjuntos de datos de alta calidad de referencia. Estos conjuntos se han utilizado previamente para la evaluación empírica e imparcial de las herramientas del estado del arte para la predicción de la actividad antimicrobiana (Gabere y Noble, 2017). Además, mostramos la capacidad de nuestra propuesta para clasificar un solo subconjunto de AMPs, estos son los péptidos antibacterianos.

El presente capítulo está organizado de la siguiente manera: en la Sección 5.1 definimos formalmente el problema que queremos resolver. En la Sección 5.2 describimos los materiales y métodos que adoptamos para resolver el problema de ponderación de características. La evaluación de nuestro enfoque y la comparación de nuestros modelos con las herramientas del estado del arte para la predicción de la actividad antimicrobiana se presenta la Sección 5.3. Por último, discutimos los principales resultados en la Sección 5.4.

5.1. Planteamiento del problema: ponderación de características

El problema general a resolver se denomina problema de ponderación de características (Hocke y Martinetz, 2015), la complejidad de este problema es referido como uno de la clase NP-Difícil (Amaldi y Kann, 1998b). Para nuestros propósitos, modelamos este problema como un problema de optimización multiobjetivo (MOP por sus siglas en inglés de *Multi-Objective Optimization*) donde se busca encontrar un conjunto de vectores de peso que simultáneamente minimicen la distancia entre AMPs y maximicen las distancias entre las clases AMPs y no AMPs.

Para definir el MOP, seguimos un enfoque similar al presentado en (Paul y Das, 2015). Las principales diferencias entre ambos enfoques son las siguientes: en primer lugar, el problema general de la ponderación en Paul y Das (2015) minimiza simultáneamente la distancia intraclase para todas las clases. En cambio, nuestro enfoque (Beltrán *et al.*, 2017) minimiza sólo la distancia intraclase de los AMPs, ya que el conjunto de no AMPs puede contener péptidos con diferentes actividades biológicas, por lo que tratar de reducir la distancia intraclase para los no AMPs sería contradictorio con el principio de similitud de propiedad (Cai *et al.*, 2013). Además, en nuestro enfoque, el número de pesos distintos de cero se utiliza como criterio de desempate para los

vectores de peso con las mismas distancias intra o inter clase (Beltran *et al.*, 2018).

A continuación, se describe formalmente el problema y los elementos importantes en el proceso de ponderación de características.

5.1.1. Notación y definiciones

Antes de presentar de manera formal la definición del problema, introducimos la notación y las definiciones necesarias.

Vector de ponderación

Dado $\mathcal{X} = \{X_1, \ldots, X_m\}$ donde $X_1, \ldots, X_m$ son las características de entrada. Llamamos a $\mathbf{w} = [w_1, \ldots, w_m]^T$ un **vector de ponderación**, con este es posible especificar el factor de escalamiento correspondiente a la i-ésima característica de la siguiente manera (Paul y Das, 2015):

$$w_i = \begin{cases} [1, \mathcal{A}] & \text{sí la característica } X_i \text{ es seleccionada;} \\ 0 & \text{si la característica } X_i \text{ es rechazada.} \end{cases} \tag{23}$$

donde $\mathcal{A} \in \mathbb{R}_{>0}$ es la ponderación máxima para w_i y puede tomar cualquier número real positivo. Tal como en (Paul y Das, 2015), consideramos $\mathcal{A} = 10$.

Escalamiento de las características

A continuación, presentamos el escalamiento de características para un vector de pesos en particular. El escalamiento está dado por la composición de matrices utilizando la multiplicación por elementos (*component-wise multiplication*).

Sea X la matriz de descriptores cuya dimensión es $n \times m$, donde n es el número de péptidos representados con m descriptores moleculares. Sea W una matriz de tamaño $n \times m$ que denota n copias de un vector de ponderación en particular, $\mathbf{w} = [w_1, \ldots, w_m]^T$. El escalamiento de las características está dado por el producto de Hadamard, también conocido como producto de Schur (Horn *et al.*, 1990), este es definido como sigue,

$$\begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix} \circ \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} = \begin{bmatrix} w_{11}x_{11} & \cdots & w_{1m}x_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1}x_{n1} & \cdots & w_{nm}x_{nm} \end{bmatrix}.$$

De esta forma en la matriz resultante, los descriptores moleculares rechazados corresponden a columnas cuyos valores son cero.

La distancia ponderada de Manhattan

Dado dos instancias $\mathbf{x}_p$ y $\mathbf{x}_q$, y un vector de ponderación $\mathbf{w}$, la distancia ponderada (también conocida como distancia ponderada de Manhattan) es definida como sigue:

$$d(\mathbf{w}, \mathbf{x}_p, \mathbf{x}_q) = \sum_{i=1}^{m} w_i |x_{pi} - x_{qi}|, \tag{24}$$

de esta manera,

$$d(\mathbf{w}, \mathbf{x}_p, \mathbf{x}_q) = \mathbf{w}^{\mathbf{T}} |\mathbf{x}_p - \mathbf{x}_q|, \tag{25}$$

donde |.| denota la norma L_1.

Distancia intraclase para la clase de interés

Dado un conjunto de datos de entrenamiento $\mathcal{D}$ y un vector de ponderación $\mathbf{w}$, suponga que la etiqueta de la clase de interés es $y = AMP$, entonces la distancia intraclase para la clase de interés está definida como sigue:

$$D_{intra}(\mathbf{w}, \mathcal{D}) = \sum_{p=1}^{n-1} \sum_{\substack{q=p+1 \\ yp, yq=AMP}}^{n} d(\mathbf{w}, \mathbf{x}_p, \mathbf{x}_q), \tag{26}$$

en donde para el cálculo de D_{intra} solo se consideran las instancias en $\mathcal{D}$ para las cuales

tienen la etiqueta de *AMP*. Dado que d es simétrica (*i.e.*, $d(\mathbf{w}, \mathbf{x}_p, \mathbf{x}_q) = d(\mathbf{w}, \mathbf{x}_q, \mathbf{x}_p)$) asignamos $q = p + 1$ en la Ecuación 26 con el objetivo de calcular solo una vez d por cada par de instancias $\mathbf{x}_p$ y $\mathbf{x}_q$.

Utilizando la definición de la distancia ponderada de la Ecuación 25, se puede reformular la distancia intraclase como sigue:

$$= \sum_{p=1}^{n-1} \sum_{\substack{q=p+1 \\ y_p, y_q = AMP}}^{n} \mathbf{w^T} |\mathbf{x}_p - \mathbf{x}_q| = \mathbf{w^T} \sum_{p=1}^{n-1} \sum_{\substack{q=p+1 \\ y_p, y_q = AMP}}^{n} |\mathbf{x}_p - \mathbf{x}_q|. \tag{27}$$

Si consideremos el último término como el vector de diferencias $\mathbf{\Delta}^{intra}$ la distancia intraclase queda reformulada como:

$$D_{intra}(\mathbf{w}, \mathcal{D}) = \mathbf{w^T} \mathbf{\Delta}^{intra}. \tag{28}$$

Una de las ventajas de reformular D_{intra} como en la Ecuación 28 es que se puede calcular la distancia intraclase para N diferentes vectores de peso con solo calcular $\mathbf{\Delta}^{intra}$ en una sola ocasión, en lugar de calcularla N veces (Paul y Das, 2015). De otra manera, calcular la función $\mathbf{\Delta}^{intra}$ tiene un tiempo de ejecución de $\mathcal{O}(n^2)$, donde n es el número de péptidos, mientras que calcular el producto punto de la ecuación 28 tiene un tiempo de ejecución de $\mathcal{O}(m)$, en donde m es el número de descriptores moleculares.

Distancia interclase

Dado un conjunto de datos de entrenamiento $\mathcal{D}$ y un vector de ponderación $\mathbf{w}$. La distancia interclase se define a continuación:

$$D_{inter}(\mathbf{w}, \mathcal{D}) = \sum_{p=1}^{n-1} \sum_{\substack{q=p+1 \\ y_p \neq y_q}}^{n} d(\mathbf{w}, \mathbf{x}_p, \mathbf{x}_q). \tag{29}$$

Utilizando la definición de la distancia ponderada de la Ecuación 25 es posible reformular la distancia interclase como sigue:

$$D_{inter}(\mathbf{w}, \mathcal{D}) = \mathbf{w^T \Delta}^{inter}, \tag{30}$$

donde $\mathbf{\Delta}^{inter}$ es el vector de las diferencias para las instancias que pertenecen a diferentes clases.

5.1.2. Enfoque multiobjetivo para el problema de ponderación de características

Dado el conjunto de entrenamiento $\mathcal{D}$ con n instancias representadas en m características de entrada, suponga que para cada instancia $\mathbf{x}_i^T \in \mathcal{D}$, el valor x_{ij} está en el intervalo $[1, \mathcal{A}]$, donde x_{ij} es la j-ésima componente del vector $\mathbf{x}_i^T$. Además, suponga que para D_{intra} la etiqueta de la clase de interés es $y = AMP$. El problema de ponderación multiobjetivo puede formularse como:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimizar}} \quad F(\mathbf{w}) = [f_1(\mathbf{w}), f_2(\mathbf{w})]^T \\ &\text{sujeto a} \qquad w_i \in \{0\} \cup [1, \mathcal{A}]\ i = 1, \ldots, m, \end{aligned} \tag{31}$$

donde,

$$f_1(\mathbf{w}) = D_{intra}(\mathbf{w}, \mathcal{D}) + \frac{[min\{1, \mathbf{w}\}]^T \mathbf{1}}{m},$$

$$f_2(\mathbf{w}) = -D_{inter}(\mathbf{w}, \mathcal{D}) + \frac{[min\{1, \mathbf{w}\}]^T \mathbf{1}}{m},$$

el término $[min\{1, \mathbf{w}\}]^T \mathbf{1}$ mide la complejidad de la solución en función del número de ponderaciones diferentes de cero, en donde min es una función que toma el menor valor de 1 y la i-ésima componente de $\mathbf{w}$, w_i para $i = 1, \ldots, m$, $\mathbf{1}$ es un vector columna de dimensión m, cuyos elementos toman valor de uno.

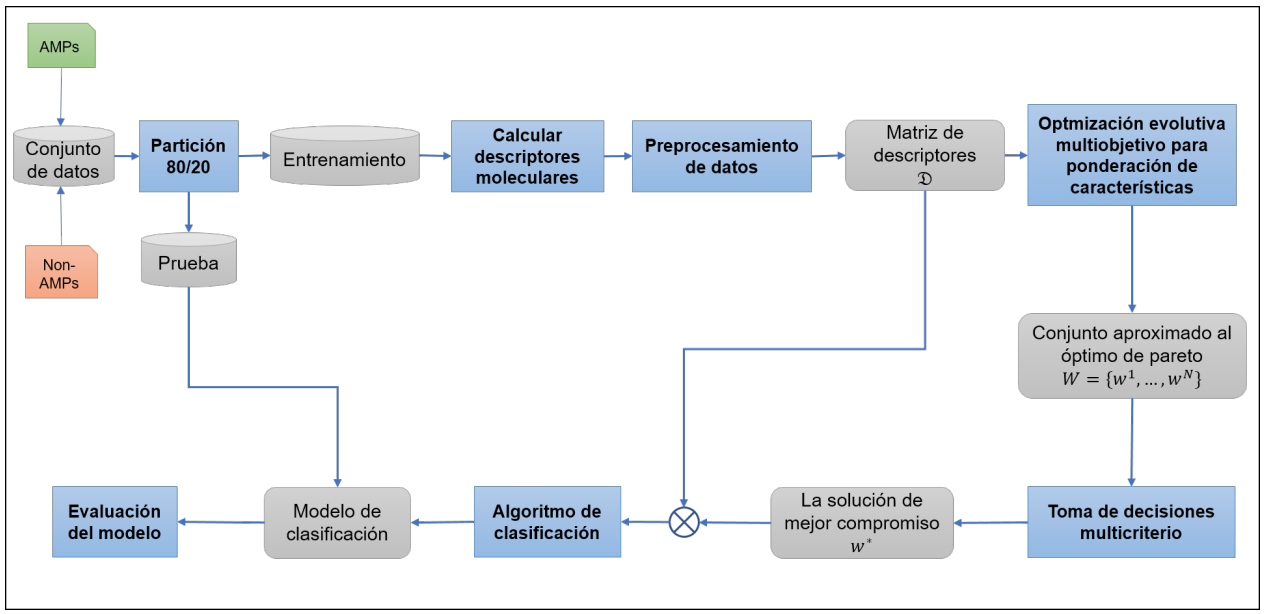


Figura 15. Esquema general para la ponderación de características. Los rectángulos con textos en negrita representan los procesos mientras que los rectángulos redondeados representan las entradas y salidas de los procesos.

5.2. Métodos

El esquema general de la metodología adoptada en este trabajo se muestra en la Figura 15. Cada proceso se describe en detalle en esta sección, incluyendo la selección y división del conjunto de datos, el cálculo y preprocesamiento de los descriptores moleculares, la ponderación de los descriptores moleculares y la clasificación de los péptidos antimicrobianos utilizando la ponderación de los descriptores como representación.

5.2.1. Conjunto de datos

Para este estudio, utilizamos los seis conjuntos de datos propuestos por (Gabere y Noble, 2017) (ver Capítulo 2, Sección 2.3.1.1). Es importante señalar que los conjuntos de datos se obtuvieron a partir de los datos suplementarios disponibles públicamente del trabajo de Gabere y Noble (2017). De estos conjunto eliminamos las secuencias que contienen residuos no estándares (*e.g.*, aminoácidos indeterminados como 'X', 'B', 'J' o 'Z'). Después, seleccionando aleatoriamente los elementos, se dividió cada conjunto de datos en dos: uno de entrenamiento y otro de pruebas. El conjunto de en-

trenamiento contiene el 80 % de las secuencias seleccionadas al azar del conjunto de datos original, mientras que el conjunto de pruebas contiene las secuencias restantes (ver Tabla 10). El conjunto de entrenamiento se utiliza en los siguientes pasos de esta sección, mientras que el conjunto de prueba sólo se utiliza para probar la efectividad de los modelos generados por nuestro enfoque (ver subsecciones 5.3.4 y 5.3.5).

Tabla 10. Resumen de los conjuntos de datos de péptidos.

Conjunto de datos	No. de secuencias AMP	No. de secuencias Non-AMP	Total
DAMPD_AMP	438	2174	2612
DAMPD_ANTIBACTERIAL	255	1242	1497
DAMP_BACTEROCIN	24	123	147
APD3_AMP	1360	6860	8220
ADP3_ANTIBACTERIAL	1158	5777	6935
ADP3_BACTEROCIN	125	612	737

* El conjunto de datos fue extraído de (Gabere y Noble, 2017) y las secuencias con aminoácidos no estándar fueron removidas.

5.2.2. Cálculo de descriptores moleculares

Para el cálculo de los descriptores moleculares se utilizaron dos paquetes de software libre: Tango (Rousseau *et al.*, 2006; Fernandez-Escamilla *et al.*, 2004; Linding *et al.*, 2004b) y la herramienta interna MODAMP (por sus siglas en inglés de MOlecular Descriptor for AntiMicrobial Peptides) (ver Capítulo 3). La primera se utilizó para calcular las siguientes propiedades fisicoquímicas: tendencia de los residuos en la secuencia a formar alfa hélices, hojas plegadas beta y agregación *in vitro*. Mientras que MODAMP (Beltrán *et al.*, 2017) se utilizó para codificar descriptores 0D y 1D. Desafortunadamente, los descriptores 3D no fueron calculados debido a la no disponibilidad de estructuras 3D para la mayoría de los AMPs conocidos. En total se calcularon cuatro descriptores moleculares usando Tango (Rousseau *et al.*, 2006; Fernandez-Escamilla *et al.*, 2004; Linding *et al.*, 2004b) y otros 268 con la herramienta MODAMP (Beltrán *et al.*, 2017). Estos descriptores se extrajeron para cada secuencia de péptidos en los conjuntos de datos de entrenamiento y prueba.

5.2.3. Preprocesamiento

Se realizó un preprocesamiento en dos niveles sobre la matriz de descriptores generada previamente. Primero, aplicamos un preprocesamiento a nivel de instancia con

el fin de eliminar los valores atípicos. Para nuestro trabajo, consideramos un valor atípico como una instancia alejada al resto de las instancias que pertenecen a la misma clase. Los valores atípicos pueden afectar el desempeño de la caracterización del espacio químico. Segundo, aplicamos un preprocesamiento a nivel de descriptor con el objetivo de poner a los valores de los descriptores moleculares en el mismo intervalo. Esto se debe a que los descriptores moleculares empleados toman valores en diferentes intervalos. Por ejemplo, el punto isoeléctrico toma valores del orden de 10^0 a 10^1 unidades de pH, mientras que el peso molecular toma valores en el orden de 10^2 a 10^3 Dalton.

Para eliminar los vectores aislados, se usó el método *Local Outlier Factor* (LOF) (Breunig *et al.*, 2000). Es importante señalar que LOF fue aplicado a cada clase (*e.g.,* AMP y no AMP) para cada conjunto de datos. En cuanto al preprocesamiento a nivel de descriptor, aplicamos el método de escalado Min-Max, que transforma los valores de cada descriptor en un intervalo entre 1 y 10 (Beltrán *et al.*, 2017).

Como resultado, obtuvimos una matriz de descriptores normalizados $\mathcal{D}$, esta matriz es la que se utiliza en los siguientes pasos de nuestro esquema general (ver Figura 15).

5.2.4. Optimización evolutiva multiobjetivo para la ponderación de características

Para resolver el problema de ponderación de características multiobjetivo (Ecuación 31) se empleó el algoritmo evolutivo multiobjetivo basado en descomposición (MOEA/DDE por sus siglas en inglés) (Zhang y Li, 2007; Li y Zhang, 2009). Li y Zhang (2009) muestran que el MOEA/D-DE funciona mejor que el bien conocido NSGAII (Deb *et al.*, 2002) para problemas de optimización continua, como el descrito en este trabajo.

En resumen, MOEA/D-DE descompone el problema de optimización multiobjetivo en N problemas de optimización mono-objetivo a través de la adopción del enfoque Tchebycheff (Miettinen, 2012). En este enfoque, el i-ésimo problema mono-objetivo o escalar es de la siguiente forma:

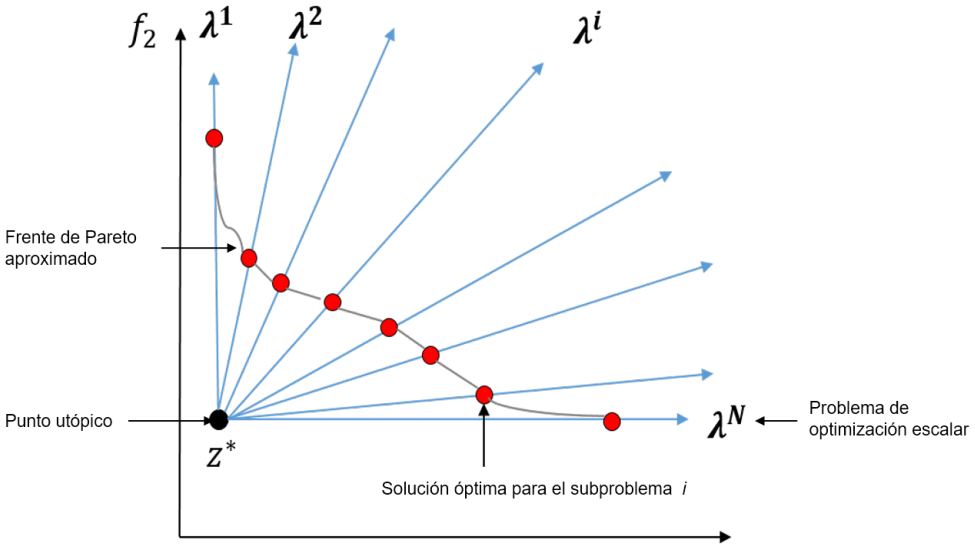


Figura 16. Ilustración de la descomposición del problema multiobjetivo en N problemas de optimización mono-objetivo.

$$\begin{aligned} &\text{minimizar} \quad g^{te}(\mathbf{w}|\boldsymbol{\lambda}, \mathbf{z}^*) = \max_{1\leq i\leq m} \{\lambda_i |f_i(w) - z_i^*|)\} \\ &\text{sujeto a} \quad \mathbf{w} \in \Omega, \end{aligned} \tag{32}$$

donde $\mathbf{z}^* = [z_1^*, \ldots, z_m^*]^T$ es el punto de referencia, también conocido como punto utópico, debido a que es una solución inalcanzable, compuesta por el mejor valor de cada objetivo, *i.e.*, $z_i^* = \text{mín}[f_i(\mathbf{w})|\mathbf{w}]$ para cada $i = 1, \ldots, m$. Ω es el espacio de decisión m-dimensional, en donde cada eje de coordenadas corresponde a una componente del vector de ponderación $\mathbf{w}$. Una ilustración de la descomposición de los problemas multiobjetivo y de los elementos que los integra se muestra en la Figura 16. Después, los N problemas mono-objetivo se resuelven simultáneamente (para una descripción detallada de este método referimos al lector interesado a (Zhang y Li, 2007; Li y Zhang, 2009)).

En general, MOEA/D-D recibe como entrada la matriz de descriptores $\mathcal{D}$ y da como salida un conjunto de N soluciones óptimas aproximadas al problema de ponderación multiobjetivo (ver problema de optimización (31)). El conjunto de soluciones encontradas es conocido como conjunto aproximado de Pareto: $\mathcal{P}^* = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$. Debe tenerse en cuenta que cada solución es un vector de peso $\mathbf{w}^k = [w_1^k, \ldots, w_m^k]^T$, donde el componente i-ésimo es el factor de escala para el descriptor molecular X_i. Para cada solución $\mathbf{w}^k$ en $\mathcal{P}^*$, un vector objetivo $F(\mathbf{w}^k) = [f_1(\mathbf{w}^k), f_2(\mathbf{w}^k)]^T$ es asignado.

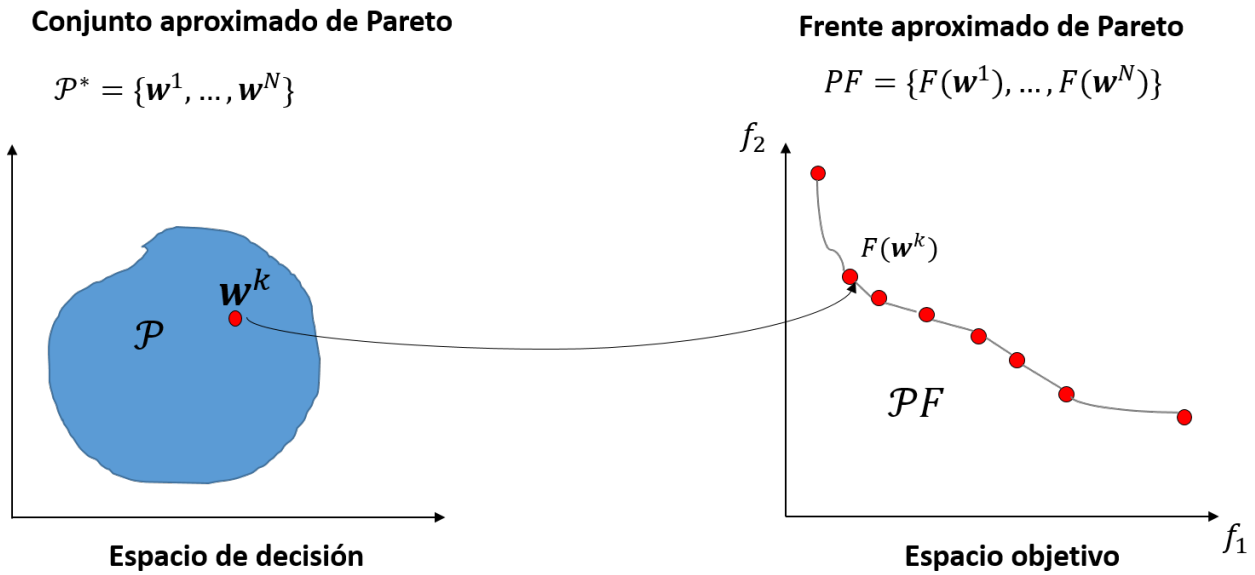


Figura 17. Ilustración de los espacios involucrados en el problema de ponderación de pesos multiobjetivo. La figura de la izquierda muestra el espacio de decisión *m*-dimensional, en este espacio coexisten todos los posibles vectores de pesos. La figura de la derecha es donde se lleva a cabo la optimización en el cuál coexisten las funciones objetivo, en este espacio cada eje de coordenadas corresponde a una componente del vector objetivo.

En este caso, el conjunto de todos estos vectores objetivos es conocido como frente aproximado de Pareto (Coello *et al.*, 2007): $PF = \{F(\mathbf{w}^1), \ldots, F(\mathbf{w}^N)\}$ (ver Figura 17).

Es importante señalar que, las soluciones en $\mathcal{P}^*$ no pueden ser consideradas mejores entre ellas en ambos objetivos ya que están en una relación de compromiso. Esto significa que, algunas soluciones en $\mathcal{P}^*$ son mejores en el objetivo f_1 que en f_2 y viceversa (ver Figura 18). Para extraer algunas soluciones de $\mathcal{P}^*$, tomando en consideración los diferentes niveles de satisfacción de las soluciones hacia los objetivos, empleamos un proceso bien establecido en la toma de decisiones multicriterio (Coello *et al.*, 2007).

5.2.4.1. Toma de decisiones multicriterio para seleccionar los vectores de pesos

Para el problema de seleccionar unos pocos vectores de peso del conjunto aproximado de Pareto $\mathcal{P}^*$ proponemos un enfoque de toma de decisiones multicriterio. Los pasos principales pueden describirse de la siguiente manera:

Paso 1: por cada solución, $\mathbf{w} \in \mathcal{P}^*$, escalar los valores para las funciones objetivo $f_1(\mathbf{w})$ y $f_2(\mathbf{w})$ en el intervalo entre [0 y 1], donde 1 significa satisfacción

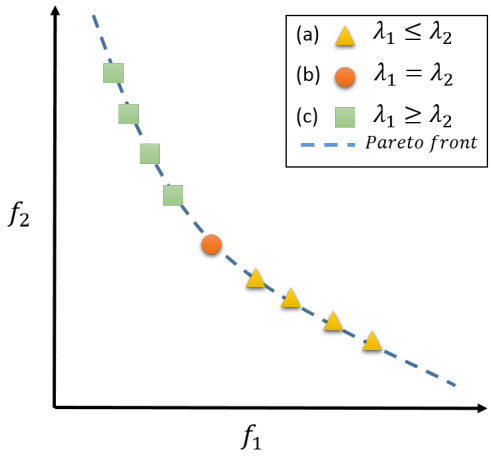


Figura 18. Ilustración del enfoque de la suma ponderada. a) f_1 es menos importante que f_2. b) f_1 es igualmente importante que f_2. c) f_2 es menos importante que f_1.

completa para un objetivo en particular y 0 indica insatisfacción. Para realizar el escalamiento utilizamos el mismo procedimiento que en Paul y Das (2015), para cada solución $\mathbf{w}^k$ y objetivo f_i realizamos el cálculo como sigue:

$$\mu_i^k = \begin{cases} 1 & \text{if } f_i(\mathbf{w}^k) = f_i^{\text{mín}}, \\ \frac{f_i^{\text{máx}} - f_i(\mathbf{w}^k)}{f_i^{\text{máx}} - f_i^{\text{mín}}} & \text{if } f_i^{\text{mín}} < f_i^k < f_i^{\text{máx}}, \\ 0 & \text{if } f_i(\mathbf{w}^k) = f_i^{\text{máx}}, \end{cases} \tag{33}$$

donde,

$$f_i^{\text{mín}} = \min_{1 \leq j \leq N} \{f_i(\mathbf{w}^j)\}, \tag{34}$$

$$f_i^{\text{máx}} = \max_{1 \leq j \leq N} \{f_i(\mathbf{w}^j)\}. \tag{35}$$

$\boldsymbol{\mu}^k = [\mu_1^k, \mu_2^k]^T$ es el vector objetivo limitado al intervalo [0,1] para la solución $\mathbf{w}^k$ en el conjunto aproximado de Pareto $\mathcal{P}^*$.

Paso 2: dado el vector de ponderación $\boldsymbol{\lambda} = [\lambda_1, \lambda_2]^T$ hacer una suma ponderada. Aquí λ_1 y λ_2 son usados para fijar las preferencias sobre los objetivos f_1 y f_2. Por ejemplo, si deseamos una solución que satisfaga a f_1 más que a f_2, entonces debe asignarse un valor mayor a λ_1 que a λ_2 (ver Figura 18).

Dado $\boldsymbol{\lambda}$, la suma ponderada g^{bcs} para cada vector objetivo $\boldsymbol{\mu}^k$ es calculada como sigue:

$$g^{bcs}(\boldsymbol{\mu}|\lambda_1) = \lambda_1\mu_1 + (1-\lambda_1)\mu_2 \tag{36}$$

Paso 3: encontrar la solución de mejor compromiso dado $\boldsymbol{\lambda}$, esto es, el vector de peso $\mathbf{w}^{k*}$ con el máximo valor de g^{bcs} (se describe formalmente en la Ecuación 37).

$$k^* = \underset{k\in[1,N]}{\arg\max} \quad g^{bcs}(\boldsymbol{\mu}^k|\lambda_1) \tag{37}$$

En este trabajo, para cada conjunto de datos, seleccionamos cinco soluciones de mejor compromiso usando λ_1 igual a 0.4, 0.45, 0.5, 0.55, y 0.60, respectivamente. Estos valores de λ_1 se seleccionaron con el fin de obtener soluciones que tengan preferencias para alguno de los dos objetivos y una solución donde ambos objetivos son igual de importantes. Para los pesos $\lambda_1 = 0.4, 0.45$ el objetivo f_1 es menos importante que f_2. Mientras que para los pesos $\lambda_1 = 0.55, 0.6$ el objetivo f_2 es menos importante que f_1. Por último, para el peso $\lambda_1 = 0.5$ ambos objetivos son igualmente importante. Además, este valor corresponde a la misma solución elegida por Paul y Das (2015).

5.2.5. Algoritmos de clasificación

El siguiente paso es escalar la matriz de descriptores $\mathcal{D}$ con las soluciones óptimas de mejor compromiso dado λ_1 (ver Subsección 5.2.4.1). Para lograr el escalamiento usamos el producto de Hadamard (definido en Subsección 5.1.1), este es aplicado a cada solución de compromiso óptima (es decir, el vector de peso $\mathbf{w}^k$) y el conjunto de datos $\mathcal{D}$, resultando en un nuevo conjunto de datos $\mathcal{D}_{\mathbf{w}}$ con la siguiente forma:

$$\hat{\mathcal{D}}_{\mathbf{w}} = \left[\begin{array}{cccc|c} w_1x_{11} & w_2x_{12} & \cdots & w_mx_{1m} & y_1 \\ w_1x_{21} & w_2x_{22} & \cdots & w_mx_{2m} & y_2 \\ \vdots & & \ddots & \vdots & \vdots \\ w_1x_{n1} & w_2x_{n2} & \cdots & w_mx_{nm} & y_n \end{array}\right]. \tag{38}$$

De esta forma, tras aplicar nuestra propuesta, los descriptores moleculares rechazados corresponden a columnas cuyos valores son cero y esas columnas fueron eliminadas. Después, con el objetivo de medir la calidad de las soluciones encontradas, construimos modelos para la clasificación binaria de la actividad antimicrobiana utilizando $\hat{\mathcal{D}}_{\mathbf{W}}$. Para la construcción de los modelos utilizamos cuatro algoritmos de aprendizaje de máquina bien conocidos: bosques aleatorios (RF), los *k* vecinos más cercanos (KNN), máquina de soporte vectorial lineal (SVM-L) y perceptrón multicapa (MLP).

En resumen, el número de clasificadores que se construyen son 20. En donde son 4 por cada solución óptima dado un valor de λ_1 ($\lambda_1 \in [0.4, 0.45, 0.5, 0.55, 0.60]$).

5.2.6. Detalles de implementación

Todos los experimentos se realizaron bajo las siguientes condiciones: sistema operativo: Ubuntu 16.04 LTS; CPU: Intel i7 a 2.40GHz; y memoria RAM: 12 GB.

El algoritmo MOEA/D-DE se implementó en Java utilizando la librería de metaheurísticas para resolver problemas de optimización multiobjetivo, *MOEA Framework 2.1* (disponible en http://www.moeaframework.org). Los parámetros principales para MOEA/D-DE se establecieron de acuerdo con los valores recomendados en (Li y Zhang, 2009) para problemas de dos objetivos, los parámetros específicos se muestran en la Tabla 11 (Beltrán *et al.*, 2017). Con ayuda de esta librería resolvemos la Subsección 5.2.4.

Para el escalamiento de la matriz de descriptores $\mathcal{D}$ con las soluciones óptimas de mejor compromiso dado λ_1 utilizamos Python 3.6. Aquí implementamos el producto de Hadamard (definido en Subsección 5.1.1) de manera independiente al programa que se implementó en Java.

Los algoritmos de aprendizaje de máquina para la clasificación se implementaron usando la librería para Python 3.6 de Scikit-learn (Pedregosa *et al.*, 2011). Scikit-learn es un conjunto eficiente de herramientas para la implementación de algoritmos de aprendizaje automático para tareas de minería de datos. Los hiperparámetros de los algoritmos de aprendizaje de máquina se resumen a continuación: KNN ($p = 1, peso = distancia$) y $k = 19, 22, 3$ para los conjuntos de datos antimicrobianos, antibacterianos y bacteriocínicos, respectivamente; SVM-L (class_weight=balanceado) y el parámetro de penalización C = 0,001, 0,1 y 0,001 para los conjuntos de datos

Tabla 11. Valores de parámetros usados en MOEA/D-DE.

Símbolo	Valores	Descripción
Parámetros de control en el cruzamiento DE y la mutación polinomial		
CR	1.0	Probabilidad de cruzamiento
F	0.5	Factor de escalamiento
η	20	Índice de distribución de la mutación polinomial
p_m	$\frac{1}{n}$	Probabilidad de mutación
Tiempo de ejecución y condición de paro		
N_{pop}	500	Tamaño de la población
N_{gen}	1000	Máximo número de generaciones
N_r	30	Número de ejecuciones del algoritmo
Parámetros de control en MOEA/D-DE		
T	20	Tamaño del vecindario
δ	0.9	La probabilidad de selección de padres del vecindario
n_r	2	Número máximo de soluciones reemplazadas por cada hijo

antimicrobianos, antibacterianos y bacteriocín, respectivamente; RF (criterion=gini, max_features=sqrt); finalmente, para el MLP utilizamos el valor predeterminado de hiperparámetros.

5.3. Resultados

Para evaluar la eficacia de nuestro enfoque, llamado MOEA-FW, realizamos experimentos con seis conjuntos de datos de alta calidad. Recientemente, estos conjuntos se han utilizado para la evaluación empírica e imparcial del estado del arte de las herramientas de predicción antimicrobiana (Gabere y Noble, 2017). Estos conjuntos de datos se seleccionaron porque están compuestos de AMPs depurados manualmente y validados experimentalmente; en estos conjuntos de datos, los no AMPs tienen la misma distribución peptídica que la observada en los AMPs (ver Sección 5.2.1).

Los experimentos realizados se pueden dividir en cuatro partes. En la primera parte, nos propusimos seleccionar los descriptores moleculares apropiados para cada conjunto de datos a través de su escalado. Mientras que, en la segunda parte, se inducen diferentes modelos de clasificación utilizando cuatro algoritmos de aprendizaje de máquina y los conjuntos de datos transformados. En la tercera parte, se utilizaron los

mejores modelos de clasificación generados para predecir la actividad antimicrobiana de nuevas secuencias de péptidos, es decir, secuencias de péptidos que no se han utilizado ni para obtener los vectores de peso ni para elegir los mejores clasificadores. Finalmente, comparamos nuestro resultado con los presentados por Gabere y Noble (2017) que evalúa diferentes predictores de AMPs.

5.3.1. Medidas de desempeño

Para comparar las soluciones de mejor compromiso encontradas por nuestro algoritmo MOEA-FW para cada conjunto de datos, se empleó un método de estimación del desempeño para evaluar la eficacia del modelo de clasificación. El método de estimación del desempeño se compone de la validación cruzada de 10 pliegues (10-*fold* CV) y diversas métricas de evaluación. En 10-fold CV, el conjunto de datos se divide en 10 subconjuntos no vacíos (*i.e.*, pliegues); cada subconjunto tiene aproximadamente el mismo tamaño. Se emplean nueve pliegues para el algoritmo de aprendizaje de máquina para inducir un clasificador, y el clasificador se prueba en el subconjunto restante, este procedimiento se repite diez veces. Además, el desempeño del clasificador se estima utilizando el valor promedio de las pruebas. Para probar el desempeño de la clasificación, se utilizaron las siguientes medidas: exactitud (Acc), coeficiente de correlación de Matthews (MCC), precisión (Prec), especificidad (Spec), sensibilidad (Sens), exactitud balanceada (BalACC) y el área bajo la curva ROC (AUC por sus siglas en inglés).

Dado que los conjuntos de datos son desbalanceados con respecto a las clases (*i.e.*, los AMPs y no AMPs no están representados por igual en los conjuntos de datos), utilizamos la exactitud balanceada y AUC para obtener una medida adecuada del rendimiento de los modelos inducidos.

5.3.2. Ponderación de los descriptores moleculares

La Figura 19 muestra el frente consolidado no dominado obtenido por nuestro enfoque (MOEA-FW) para cada conjunto de datos. El frente consolidado no dominado se genera después de ejecutar 30 veces MOEA-FW. Los indicadores diamante y cuadrado (*i.e.*, $\lambda_1 = 0,55$ y $\lambda_1 = 0,6$) representan las soluciones de mejor compromiso que favorecen el objetivo f_1 (*i.e.,* minimizan la distancia entre AMPs). Alternativamente, $\lambda_1 = 0,45$ y $\lambda_1 = 0,4$ representan las soluciones de mejor compromiso que dan mayor

importancia al objetivo f_2 (es decir, maximizan la distancia entre AMPs y no AMPs). Además, $\lambda_1 = 0.5$ representa el valor para la solución de mejor compromiso donde ambos objetivos son igualmente importantes.

El porcentaje de reducción del número de descriptores moleculares se muestra en la Figura 20. Los resultados muestran que la reducción en el número de descriptores moleculares por solución óptima es similar en los seis conjuntos de datos. En particular, la solución de mejor compromiso $\lambda_1 = 0.5$ tiene, en promedio, una reducción en el número de descriptores moleculares del 52.7 %, es decir, en promedio, la solución mejor ponderada tiene 128 características de un total de 272. Sin embargo, el conjunto de datos DAMP_BACTERIOCIN muestra un incremento de esta medida para la solución $\lambda_1 = 0.45$. Estos hallazgos indican que las soluciones que apoyan al objetivo f_1 (distancia entre clases) tienen, en promedio, menos descriptores moleculares que aquellas que favorecen el objetivo f_2 (distancia entre clases).

5.3.3. Selección del modelo

A continuación, por cada solución de mejor compromiso (*i.e.,* un vector de peso **w**), se transformaron los conjuntos de datos originales. Luego, para cada conjunto de datos transformado, se construyeron cuatro modelos de clasificación mediante los siguientes algoritmos de aprendizaje automático: RF, K-NN, MLP y SVM-L.

Como se mencionó, la exactitud balanceada (*BalAcc*) fue considerada como una medida para determinar el mejor modelo en los seis conjuntos de datos ponderados por las soluciones de mejor compromiso. Aplicamos la prueba no paramétrica de Friedman (Friedman, 1940) y la prueba de Nemenyi (Demšar, 2006) con el objetivo de verificar si existen diferencias significativas entre el desempeño de los clasificadores. Las pruebas de Firedman (Friedman, 1940) y Nemenyi (Demšar, 2006) han sido ampliamente usadas en la literatura para la comparación estadística de clasificadores en múltiples conjuntos de datos (el lector interesado es referido a (Demšar, 2006) para más información sobre cómo realizar ambas pruebas).

Nuestros resultados indicaron que la solución de mejor compromiso, con $\lambda_1 = 0.5$, permite inducir en promedio mejores modelos de clasificación sin importar el algoritmo de aprendizaje, el *BalAcc* fue de 87.52 %.

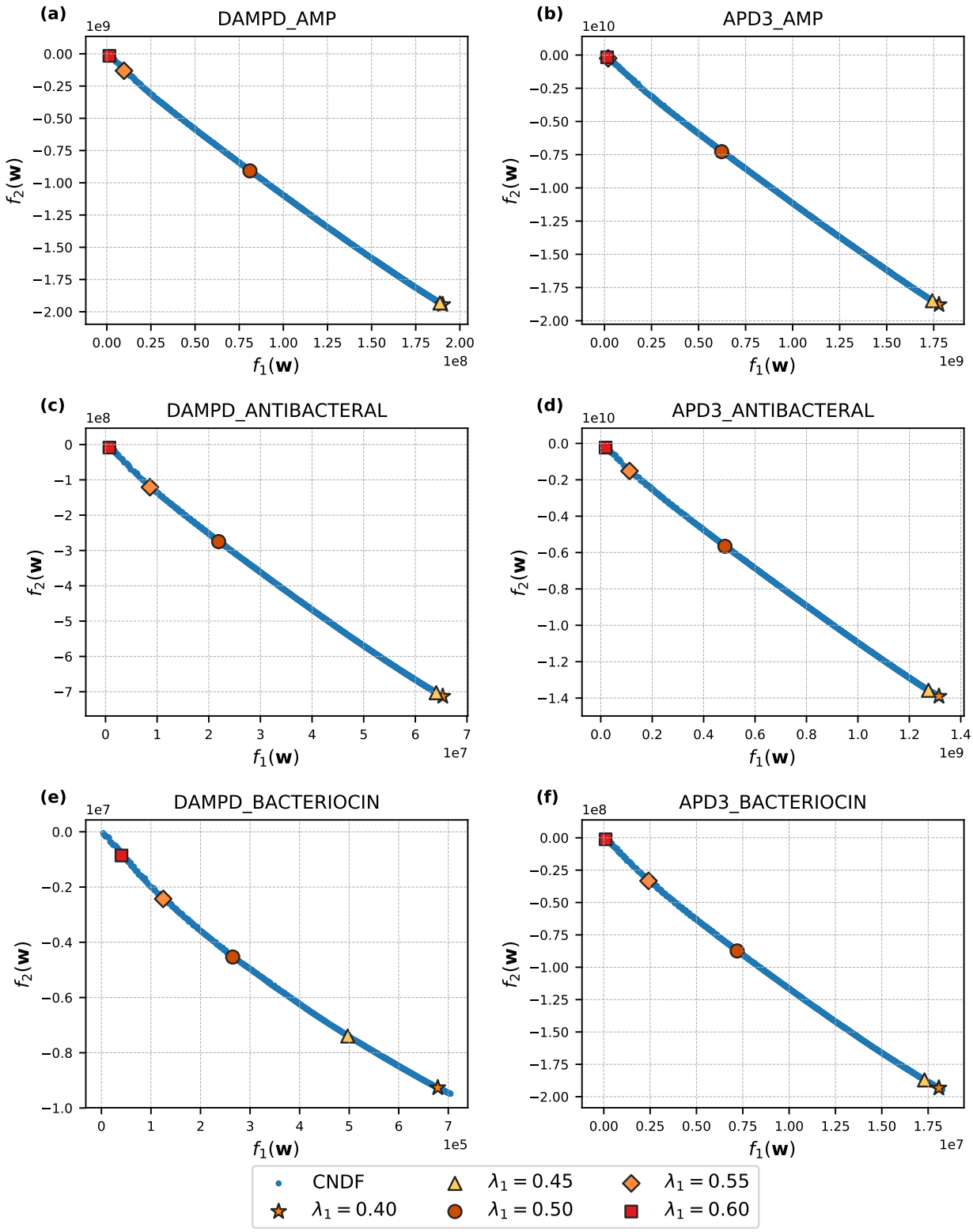


Figura 19. Visualización del frente consolidado no dominado (CNDF). El CNDF se genera después de 30 ejecuciones de MOEA-FW para cada conjunto de datos. Los marcadores representan los valores para las soluciones de mejor compromiso dado λ_1.

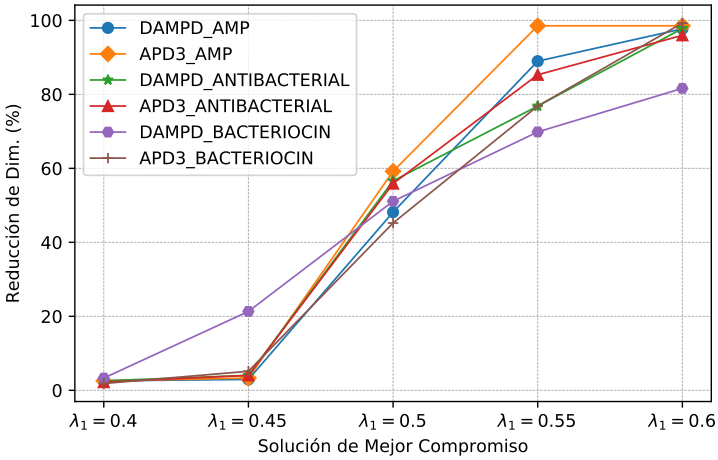


Figura 20. Porcentaje de reducción del número de descriptores moleculares para las soluciones de mejor compromiso en seis conjuntos de datos.

El análisis estadístico del desempeño de los MLAs identificó (por la prueba Friedman) una diferencia significativa en el *BalAcc* ($\chi_f^2(3) = 55.2$, valor p = 6.224e-12) de los cuatro MLAs en múltiples conjuntos de datos. Nuestros resultados muestran que, en promedio, el SVM-L ocupó el primer lugar (con rango 1.23), KNN el segundo (con rango 2.43), RF el tercero (2.63) y MLP el cuarto (3.7). Además, encontramos que el SVM-L tuvo un desempeño significativamente mejor que el MLP (Nemenyi: z = 7.4, valor p = 8.40e-13), RF (Nemenyi: z = 4.2, valor p = 0.00016), y KNN (Nemenyi: z = 3.6, valor p=0.00181). De manera similar, KNN se desempeñó significativamente mejor que MLP (z = 3,8, valor p = 0,00083). Aunque el KNN funciona un poco mejor que el RF, no hubo una diferencia estadísticamente significativa (valor p=0.932) entre ambos.

En particular, considerando sólo la solución de mejor compromiso con $\lambda_1 = 0.5$, el promedio de *BalAcc* para SVM-L fue 92.65 % y para KNN 90.13 %. Por lo tanto, nuestros hallazgos indican que para los seis conjuntos de datos, la solución de mejor compromiso con $\lambda_1 = 0.5$ usando SVM-L y KNN indujo mejores modelos de clasificación.

La Tabla 12 resume el resultado obtenido por SVM-L y KNN con la solución de mejor compromiso para $\lambda_1 = 0.5$. Los valores de la métrica representan el promedio de la validación cruzada de 10 pliegues. En esta tabla, también se realizó una prueba de Wilcoxon sobre las diferencias observadas entre KNN y SVM-L para los valores de Sens(%), Spec(%), Prec(%), BalAcc, Acc(%), MCC y AUC; si la diferencia es estadística-

mente significativa, a un nivel de confianza del 95 %, entonces se agrega un asterisco al valor ganador (en negrita). En la mayoría de los casos, los modelos de clasificación generados por el KNN muestran una mejor especificidad y precisión que los generados por el SVM-L, es decir, los modelos predicen correctamente el 96 % de los no AMPs, y clasifican correctamente el 80 % de los AMPs previstos. En contraste, el modelo de clasificación obtenido por el SVM-L mostró una buena sensibilidad, es decir, el modelo clasifica correctamente el 88.33 % de los AMPs.

Para determinar el efecto de MOEA-FW en la eficiencia de los modelos de clasificación, comparamos el rendimiento de dos clasificadores generados por el mismo algoritmo de aprendizaje de máquina, uno aplicando el MOEA-FW y el otro, utilizando todas las características de entrada (*i.e.*, línea base). Seleccionamos el mejor algoritmo de aprendizaje de máquina por base de datos, de acuerdo con la columna de exactitud equilibrada de la Tabla 12. Hicimos la prueba de Wilcoxon con el valor de *BalAcc* resultante de la validación cruzada de nuestro método y la línea de base para cada conjunto de datos. Los modelos generados por MOEA-FW muestran una mejora significativa con respecto a los modelos de línea de base sobre el *BalAcc*. Para cada conjunto de datos, la diferencia significativa en *BalAcc* entre MOEA-FW y la línea de base fue la siguiente: DAMPD_AMP (p-value = 0.00976), APD3_AMP (p-value = 0.00195), DAMPD_ANTIBACTERIAL (p-value = 0.00976), APD3_ANTIBACTERIAL (p-value = 0.00976), DAMPD_BACTERIOCIN (p-value = 0.051), y APD3_ BACTERIOCIN (p-value = 0.08398). De la misma manera, se observaron resultados similares para las otras métricas (ver Figura 21). En esta figura, un asterisco indica que la diferencia observada es estadísticamente significativa.

Por otro lado, si tomamos en consideración otras métricas (Sens, Spec, AUC, MCC, Pres, Acc) para comparar ambos modelos, los resultados muestran que los modelos generados por MOEA-FW alcanzan un rendimiento comparable o superior a los obtenidos usando todas las características de entrada. En particular, MOEA-FW logra una mejora sobre la línea de base para los conjuntos de datos DAMPD_AMP, APD3_AMP y APD3_ANTIBACTERIAL (ver Figura 21). En contraste, para los conjuntos de BACTERIOCIN, MOEA-FW muestra una disminución en la medida de exactitud con respecto a la línea base. Este último resultado sugiere que nuestra propuesta no puede encontrar un espacio químico adecuado para los conjuntos de datos de BACTERIOCIN, por lo que

Tabla 12. Desempeño de la validación cruzada de 10 pliegues en seis conjuntos de datos para KNN y SVM-L, $\lambda_1 = 0.5$.

Conjunto de datos	MLA	Sens(%)	Spec(%)	Prec(%)	Bal Acc(%)	Acc(%)	MCC	AUC
DAMPD_AMP	KNN	71.97	**97.22***	**83.75***	84.60	**93.01**	**0.735**	0.846
	SVM-L	**88.07**^{*b}	92.30	69.56	**90.19***	91.62	0.734	**0.902***
APD3_AMP	KNN	80.85	**95.27***	**77.23***	88.06	**92.85**	0.747	0.881
	SVM-L	**91.65***	92.53	70.75	**92.09***	92.36	**0.762**	**0.921***
DAMPD_ANTIBACTERIAL	KNN	**91.04**	96.45	**84.37**	**93.75**	**95.51**	**0.849**	**0.937**
	SVM-L	88.49	**96.54**	84.18	92.51	95.06	0.832	0.925
APD3_ANTIBACTERIAL	KNN	79.32	**95.30***	**77.18***	87.31	**92.61**	0.738	0.873
	SVM-L	**91.34***	92.22	70.33	**91.78***	92.07	**0.756**	**0.918***
DAMPD_BACTEROCIN	KNN	100	95.53	85.83	97.76	96.36	0.902	0.978
	SVM-L	100	**98.89**	**96.67**	**99.44**	**99.09**	**0.977**	**0.994**
APD3_BACTEROCIN	KNN	83.50	**95.04**	77.05	89.27	93.12	0.758	0.893
	SVM-L	**85.38**	94.83	**77.28**	**90.10**	93.12	**0.768**	**0.901**

Cada valor es el desempeño promedio de la validación cruzada de 10 pliegues por el clasificador construido por el algoritmo de aprendizaje de la máquina (segunda columna) en el conjunto de datos (primera columna). Se realizó una prueba de Wilcoxon sobre la medida resultante de la validación cruzada de 10 pliegues de KNN y SVM-L. Los modelos con una mejora significativa en el valor de $p \leq 0.05$ están marcados con el símbolo *.

[b] Las letras en negrita indican el mejor valor por medida para cada conjunto de datos.

con esta representación no podría inducirse un modelo eficiente para discriminar lo que es una bacteriocina. Las conjeturas del porqué está sucediendo esto se dan en la sección de Discusión 5.4.

5.3.4. Evaluación del modelo

Después de seleccionar los mejores modelos obtenidos con la solución de mejor compromiso dada $\lambda_1 = 0.5$, y utilizando los algoritmos de aprendizaje de máquina KNN y SVM-L, medimos su capacidad de predicción sobre nuevas secuencias de péptidos, es decir, secuencias de péptidos que no han sido utilizadas ni para la obtención de los vectores de peso ni para pruebas de validación cruzada (ver Sección 5.2). Observamos que todos los clasificadores inducidos por el SVM-L tienen un valor de AUC > 0.83, esto significa que los modelos generados por el SVM-L tienen una excelente capacidad para aprender qué es un péptido antimicrobiano. Mientras que el modelo generado por KNN mantiene una excelente especificidad (como indican los resultados presentados en la Tabla 12).

Por otro lado, comparando los resultados del conjunto DAMP_BACTIBASE, especialmente para la bacteriocina, en las tablas 12 y 13, la diferencia considerable de la sensibilidad (Sens(%)) puede deberse al reducido número de bacteriocinas en el conjunto

de prueba.

5.3.5. Comparación con los clasificadores de AMP existentes.

El mejor modelo generado por nuestro enfoque MOEA-FW fue comparado con otros predictores AMP, estos predictores utilizaron los mismos conjuntos de datos que los presentados en la Sección 5.2.1. Es importante notar que las cantidades de instancias entre nuestra prueba y la mostrada en (Gabere y Noble, 2017) son diferentes, dado que en (Gabere y Noble, 2017) la evaluación de los predictores de AMPs fue utlizando todas las instancias, mientras que en nuestro método utilizamos solo el 20 % para pruebas y el otro 80 % del conjunto fue usado en el proceso de optimización (ver Métodos 5.2.1). Sin embargo, esta comparación pretende mostrar la capacidad predictiva de los modelos de clasificación generados con nuestro enfoque y los presentados por los métodos del estado del arte.

El desempeño de los clasificadores para la predicción de AMPs se resume en las tablas 14 y 15. Los resultados reflejan que los modelos producidos por nuestro enfoque presentan un mejor rendimiento que los métodos del estado del arte para la clasificación de la actividad antimicrobiana y antibacteriana. Cabe destacar que los modelos derivados de nuestro enfoque para clasificar los péptidos antibacterianos superaron el desempeño de AntiBP (Lata *et al.*, 2007) y AntiBP2 (Lata *et al.*, 2010) (ver tablas 14 y 15). Sin embargo, nuestro modelo es mejorado por BAGEL3 (van Heel *et al.*, 2013) para los conjuntos de datos BACTERIOCIN.

5.4. Discusión

El enfoque propuesto pretende identificar un peso para cada descriptor molecular, de tal manera que los péptidos con actividad antimicrobiana tienden a estar muy cercanos, mientras que los péptidos con diferentes actividades biológicas tienden a estar muy alejados unos de otros. Nuestros resultados indican que la solución de mejor compromiso con $\lambda_1 = 0.5$ permite, en promedio, la mejor exactitud balanceada para las seis bases de datos. Además, esta solución permite una reducción de al menos un 52 % en el número de descriptores moleculares. Es importante destacar que un análisis preliminar (Beltrán *et al.*, 2017), en donde se evaluó la prueba de concepto de nuestra propuesta, la mejor solución, para una base de datos más pequeña, se encontró con

$\lambda_1 = 0.55$, y redujo el número de descriptores en un 67.90 %. La diferencia puede ser consecuencia de tener conjuntos de datos desequilibrados en este caso. Con la mejor solución de compromiso ($\lambda_1 = 0.5$), transformamos (ponderamos las características) los conjuntos de datos y construimos modelos para la clasificación binaria de AMPs y no AMPs. Los resultados indican que tanto el KNN como el SVM-L permiten obtener modelos eficaces para la clasificación de péptidos antimicrobianos y antibacterianos. Estos resultados apoyan la idea de que nuestro enfoque MOEA-FW permite generar mejores modelos para una actividad antimicrobiana específica, en este caso en particular, la actividad antibacteriana. En este sentido, esperamos utilizar este enfoque en el futuro para clasificar otras actividades antimicrobianas específicas, como antivirales, antifúngicas y antiparasitarias, para determinar si esta clasificación también se observa en esas actividades antimicrobianas en particular.

Como se mencionó anteriormente, los modelos generados por KNN alcanzan una alta especificidad y precisión, mientras que los modelos inducidos por SVM-L producen una alta sensibilidad (ver Tabla 12). Estos resultados sugieren que, combinando los modelos generados por KNN y SVM-L, podríamos explotar sus propiedades para generar modelos aún más eficientes.

Por otro lado, el modelo de menor rendimiento generado por MOEA-FW fue para la clasificación de péptidos cuyo origen y destino son bacterias (*i.e.,* bacteriocinas). En este caso, nuestro enfoque no fue capaz de producir un espacio químico donde tanto la actividad peptídica como su fuente pudieran ser discriminadas. Es importante señalar que BAGEL3 (van Heel *et al.*, 2013) y BACTIBASE (Hammami *et al.*, 2007) utilizan propiedades relacionadas con la similitud de secuencias para clasificar las bacteriocinas.

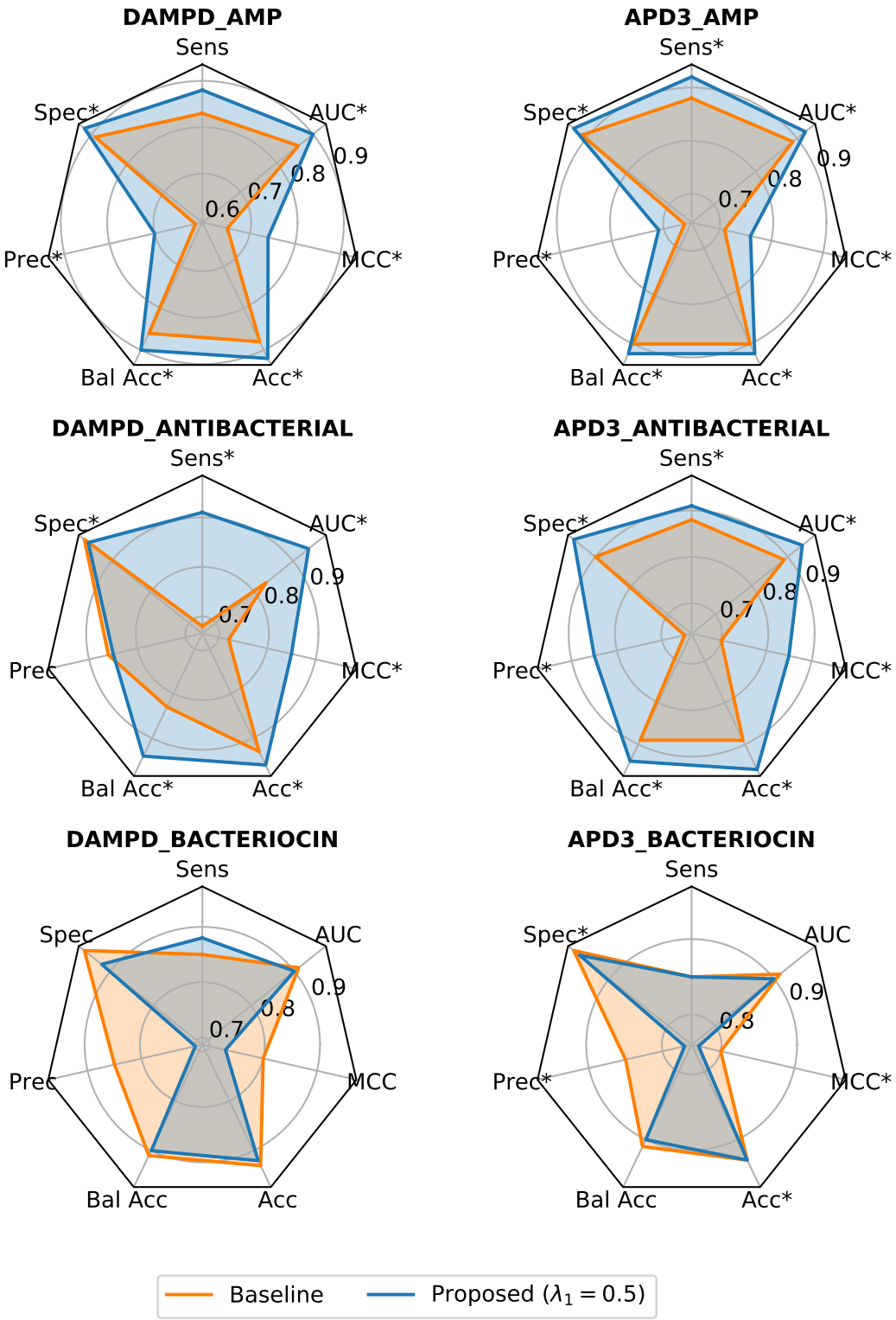


Figura 21. Comparación de desempeño entre el mejor modelo logrado por MOEA-FW y la línea de base. Cada gráfica muestra la medida de desempeño por validación cruzada de 10 pliegues para el mejor modelo logrado por MOEA-FW y la línea de base. (*i.e.*, todas las características de entrada) para un conjunto de datos en particular. El polígono representa el desempeño de todas las métricas de un modelo de clasificación en particular. Cuando un polígono está cubierto significa que el modelo es peor en todas las métricas que el modelo representado por el polígono que lo incluye. Se realizó una prueba de rangos con signo de Wilcoxon entre el mejor modelo logrado por MOEA-FW y la línea base. Los modelos con una mejora significativa en el valor de $p \leq 0.05$ están marcados con el símbolo *.

Tabla 13. Comparativa de desempeño de KNN y SVM-L en secuencias de péptidos no vistas de los seis conjuntos de datos, $\lambda_1 = 0.5$.

Conjunto de datos	ML	Sens(%)	Spec(%)	Prec(%)	Bal Acc(%)	Acc(%)	MCC	AUC
DAMPD_AMP	KNN	72.16	**94.17**	**68.63**	83.17	**90.87**	**0.650**	0.832
	SVM-L	**77.32**[b]	91.62	61.98	**84.47**	89.47	0.631	**0.845**
APD3_AMP	KNN	70.82	**92.11**	**65.10**	81.47	**88.45**	**0.609**	0.815
	SVM-L	**89.24**	82.87	51.98	**86.05**	83.97	0.597	**0.861**
DAMPD_ANTIBACTERIAL	KNN	**80.0**	90.91	60.27	**85.45**	89.30	0.634	**0.855**
	SVML	74.55	**93.10**	**65.08**	83.82	**90.37**	**0.640**	0.838
APD3_ANTIBACTERIAL	KNN	65.97	**93.91**	**68.35**	79.94	89.26	0.607	0.799
	SVM-L	**81.94**	91.55	65.92	**86.75**	**89.95**	**0.676**	**0.867**
DAMPD_BACTEROCIN	KNN	**80**	87.50	50.00	**83.75**	86.49	0.561	**0.838**
	SVM-L	60	**96.88**	**75.00**	78.44	**91.89**	**0.626**	0.784
APD3_BACTEROCIN	KNN	75.86	**94.23**	70.97	85.05	91.35	0.682	0.850
	SVM-L	**93.10**	92.95	**71.05**	**93.03**	**92.97**	**0.774**	**0.930**

* Cada valor representa el desempeño del clasificador inducido por el algoritmo de aprendizaje de máquina (segunda columna) sobre el conjunto de datos de prueba (primera columna). Al conjunto de datos de prueba se le aplicó la solución de mejor compromiso para $\lambda_1 = 0.5$.

[b] La fuente en negrita indica el mejor valor por medida para cada conjunto de datos.

Tabla 14. Comparación de desempeño entre los métodos de predicción de AMPs reportados en (Gabere y Noble, 2017) y nuestro enfoque propuesto para el conjunto de datos DAMPD.

Herramienta	Tarea	Sens(%)	Spec(%)	Prec(%)	Bal Acc(%)
MOEA-FW(SVM-L)	Antimicrobial	77.32	91.62	**61.98**	**84.47**
CAMPR3(RF)		**92.32**[b]	72.65	40.30	82.49
CAMPR3(SVM)		90.13	72.10	39.25	81.11
ADAM		84.09	68.88	35.09	76.49
MLAMP		63.62	82.27	41.78	72.94
DBAASP		22.12	**92.87**	38.28	57.49
AMPA		48.81	84.79	39.09	66.80
MOEA-FW(KNN)	Antibacterial	80.00	**90.91**	**60.27**	**85.45**
AntiBP		**89.78**	45.05	24.63	67.41
AntiBP2		86.90	15.97	17.14	51.44
MOEA-FW(KNN)	Bacteriocin	80.00	87.50	50.00	83.75
BAGEL3		**93.55**	**100.0**	**100.0**	**96.77**
BACTIBASE		83.87	**100.0**	**100.0**	91.93

[b] La letra en negrita indica el mejor valor por medida

Tabla 15. Comparación de desempeño entre los métodos de predicción de AMPs reportados en (Gabere y Noble, 2017) y nuestro enfoque para el conjunto de datos APD3.

Tool	Task	Sens(%)	Spec(%)	Prec(%)	Bal Acc(%)
MOEA-FW(SVM-L)	Antimicrobial	89.24	82.87	**51.98**	**86.05**
CAMPR3(RF)		**94.80**[b]	72.65	40.30	82.49
CAMPR3(SVM)		90.60	72.10	39.25	81.11
ADAM		91.07	68.88	35.09	76.49
MLAMP		75.59	82.27	41.78	72.94
DBAASP		62.81	92.87	38.28	57.49
AMPA		39.17	**84.79**	39.09	66.80
MOEA-FW(SVM-L)	Antibacterial	**81.94**	**91.55**	**65.92**	**86.75**
AntiBP2		66.59	26.00	15.25	46.30
MOEA-FW(SVM-L)	Bacteriocin	**93.10**	92.95	71.05	93.03
BAGEL3		86.36	**100.0**	**100.0**	**93.18**
BACTIBASE		38.36	**100.0**	**100.0**	69.48

[b] La letra en negrita indica el mejor valor por medida

Capítulo 6. Aprendizaje profundo para la clasificación y el diseño de péptidos antimicrobianos

Hoy en día, los enfoques de Aprendizaje Profundo (DL por sus siglas en inglés para *Deep Learning*) han tenido éxito en el diseño *de novo* de pequeñas moléculas, esto debido a su capacidad de buscar en regiones no exploradas del espacio químico para identificar un conjunto de moléculas *de novo* que probablemente sean activas (Sanchez-Lengeling *et al.*, 2017; Segler *et al.*, 2017). Ejemplo de estos enfoques incluyen las redes generativas adversarias (*Generative Adversaial Networks*) (Sanchez-Lengeling *et al.*, 2017), redes neuronales recurrentes (Segler *et al.*, 2017) y aprendizaje por refuerzo (Olivecrona *et al.*, 2017).

Uno de los objetivos de este trabajo es generar nuevos AMPs que presenten las siguientes características: (i) que sean secuencias de AMPs no similares a un conjunto de AMPs de entrada; (ii) que sean similares al conjunto de entrada en el espacio químico. Para lograr esto, proponemos dos modelos basados en DL: el primero es un modelo generativo para el diseño de péptidos a nivel de aminoácidos, en específico utilizamos una red neuronal bidireccional recurrente de memoria a largo y corto plazo; el segundo modelo es de clasificación binaria y tiene como objetivo determinar si los péptidos diseñados son antimicrobianos o no. Para construir este modelo utilizamos una red neuronal óptima profunda. Para el entrenamiento de ambas redes recolectamos un gran número de secuencias de péptidos, en específico para el entrenamiento de la DNN la colección de secuencias es la más grande hasta ahora utilizada para la clasificación de AMPs. La colección tiene 43403 secuencias para el entrenamiento y 8681 para las pruebas.

6.1. Métodos

El esquema general de nuestro enfoque para el diseño de nuevos AMPs se muestra en la Figura 22. El enfoque se compone de cuatro componentes principales: primero, un modelo generativo para crear secuencias de péptidos; segundo, un modelo de predicción antimicrobiana para determinar la probabilidad de que una secuencia dada sea antimicrobiana y eliminar péptidos inactivos; tercero, una función de evaluación de similitud, esta tiene como objetivo determinar el porcentaje de similitud de las nuevas

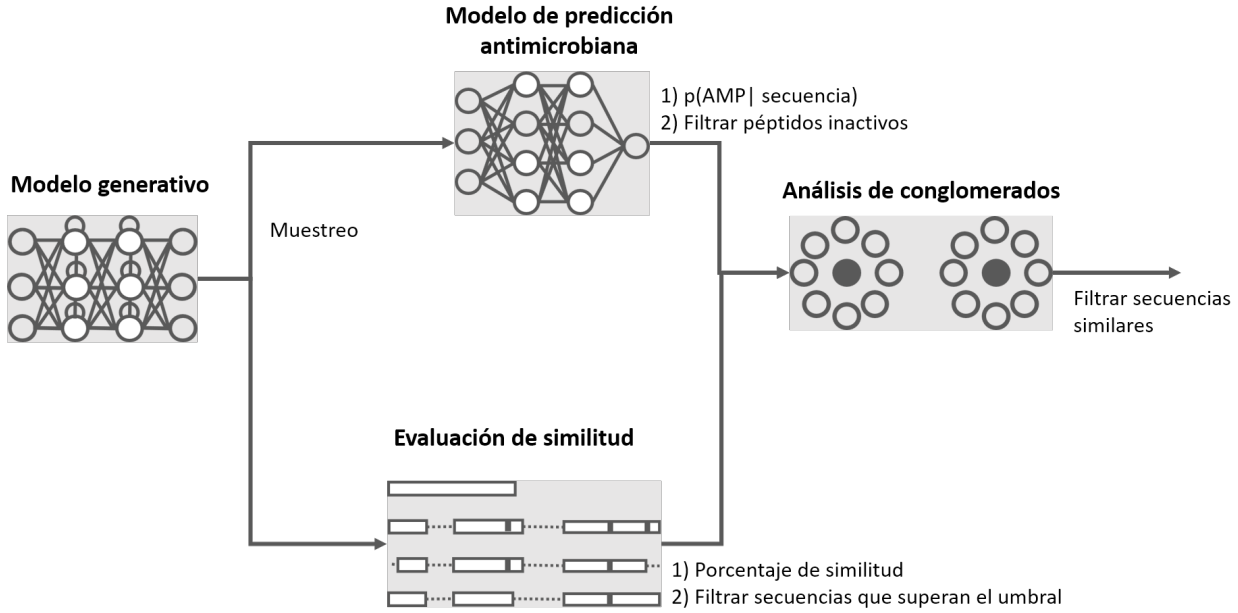


Figura 22. Esquema general para el diseño de nuevos AMPs. El enfoque está compuesto de cuatro bloques: un modelo generativo para crear nuevos péptidos; un modelo de clasificación para eliminar péptidos inactivos; una evaluación de homología para filtrar secuencias homólogas; por último, un análisis de conglomerados para las secuencias diseñadas con el objetivo de seleccionar secuencias distintas.

secuencias con respecto a las secuencias de AMPs conocidas y después eliminar las secuencias que superan un cierto umbral de similitud con los péptidos de referencia. El último componente es un análisis de conglomerados, este tiene como objetivo agrupar las secuencias que son similares dado un umbral de similitud y elegir un representante por conglomerado. El propósito de esto último es seleccionar distintas secuencias para probarlas en el laboratorio. En las siguientes subsecciones se describen los materiales y métodos necesarios para implementar los componentes.

6.1.1. Colección de datos

Recolectamos 52080 secuencias de péptidos que contienen 29400 péptidos antimicrobianos validados experimentalmente y 22688 secuencias de péptidos potencialmente no antimicrobianas. Creamos el conjunto de datos de entrenamiento utilizando el 83 % de las secuencias peptídicas totales y el conjunto de datos de prueba contiene las secuencias restantes. A continuación, se ofrece una descripción detallada de la metodología para la preparación de las secuencias AMP y no AMP.

6.1.1.1. Conjunto de AMPs

Las secuencias de AMPs fueron extraídas de StarPepDB (Aguilera-Mendoza *et al.*, 2019). StarPepDB es una base de datos Neo4j orientada a grafos, la cual es resultado de la integración, limpieza y estandarización de un gran número de bases de datos de AMPs. De StarPepDB, recuperamos 31406 AMPs validados experimentalmente, estas secuencias tienen diferentes actividades biológicas anotadas, tales como: antibacterianos (Gram+, Gram-), anti-*biofilm*, antivirales, anti-VIH, antifúngicas, antiparasitarios, antimaláricos, antiprotista, anticancerígenos, antioxidantes, quimiotácticos, insecticidas, espermicidas, inmovilizadas en la superficie y cicatrización de heridas. De estas secuencias, eliminamos 4891 secuencias de péptidos que contienen residuos no estándar (por ejemplo, aminoácidos indeterminados como X, B, J o Z). El conjunto de datos final de AMPs contiene 29400 secuencias con una longitud de 5 a 100 aminoácidos.

6.1.1.2. Conjunto de no AMPs

Crear un conjunto de datos para la clase negativa de péptidos antimicrobianos es una tarea difícil, principalmente porque las bases de datos de proteínas contienen solo información anotada sobre la evidencia experimental de la respuesta o de la actividad que tiene un péptido hacia un objetivo, mientras que los objetivos para los cuales el péptido no tiene actividad no se encuentra anotado o no se ha experimentado. Además, la ausencia de la anotación de la actividad de un péptido a un objetivo particular no garantiza la no actividad del péptido. Por otro lado, podría pensarse que una forma sencilla y segura de generar secuencias negativas es utilizando la generación aleatoria. En la generación aleatoria se supone que para cada posición de la secuencia de longitud l, cada uno de los 20 aminoácidos tiene la misma probabilidad de ocurrir. Sin embargo, los estudios de investigación han demostrado que las secuencias aleatorias con bioactividad no son raras (Neme *et al.*, 2017). Investigaciones anteriores en la clasificación de AMPs han diseñado metodologías para seleccionar secuencias que potencialmente no actúen como péptidos antimicrobianos. El uso del término “potencial” se debe a que no hay garantía de que el péptido no tenga la actividad antimicrobiana hasta su evaluación en el laboratorio, incluso después de esta evaluación la certeza

Tabla 16. Metodología empleada de cada conjunto de datos de péptidos no antimicrobianos recolectado en este trabajo.

Conjunto de datos	Base de datos de entrada	Filtro				Núm. de secuencias	Predicción
		Localización subcelular	Sesgo de homología (punto de corte)	Sin anotación	Longitud del péptido		
NANTIBP DLAMP18 (Veltri *et al.*, 2018)	Uniprot	Citoplasma	40 %	Antimicrobiano Antibiótico Antiviral Antifúngico Efector o excretado	[11, 175]	1778	Actividad antibacteriana
NAMP IAMP13 (Xiao *et al.*, 2013)	UniProt (release 2012)	N/A	40 %	Antimicrobial Antibiótico Antifúngico (Fungicida) Defensina	[5, 100]	2360	Actividad antimicrobiana
NANTIBP AntiBP07 (Lata *et al.*, 2007)	N/A	Proteínas no secretoras	N/A	N/A	[30]	431	Actividad antibacteriana
NAMP CAMP14 (Waghu *et al.*, 2014)	UniProt Secuencias arbitrarias no AMPs experimentales	N/A	90 %	Antimicrobial	[11, 80]	4011	Actividad antimicrobiana
NAMP ANFIS12 (Fernandes *et al.*, 2012)	PDB	Proteínas no membranales Proteínas no secretoras	50 %	No ARN No ADN No mezclado	[11, 100]	116	Actividad antimicrobiana
NAMP Gabere17 DAMPD (Gabere y Noble, 2017)	UniProt (2016_04)	Golgi Cytoplasm Endoplasmic reticulum mitochondria	N/A	Antimicrobiano	[11, 710]	2735	Actividad antimicrobiana
NANTIBP Gabere17_DAMPD (Gabere y Noble, 2017)					[12, 204]	1565	Actividad antibacteriana
NBactibase Gabere17_DAMPD (Gabere y Noble, 2017)					[22, 74]	155	Bacteriocina
NAMP Gabere17_APD3 (Gabere y Noble, 2017)	UniProt (2016_12)	N/A	N/A	Antimicrobiano Revisado: sí	[11, 174]	8565	Actividad antimicrobiana
NANTIBP Gabere17_APD3 (Gabere y Noble, 2017)					[11, 149]	7230	Actividad antibacteriana
NBactibase Gabere17_APD3 (Gabere y Noble, 2017)					[12, 121]	770	Bacteriocina

no es del 100 % debido a que la diversidad de microbios a ensayar es muy grande y experimentalmente se podrán ensayar solo unas cuantas.

En este trabajo, analizamos estas metodologías y sus secuencias para obtener un conjunto de datos no redundante de péptidos no antimicrobianos. En primer lugar, recogimos once conjuntos de datos de péptidos con actividad potencial no antimicrobiana, estos fueron recuperados de los trabajos de: Veltri *et al.* (2018); Gabere y Noble (2017); Xiao *et al.* (2013); Waghu *et al.* (2014); Fernandes *et al.* (2012); Lata *et al.* (2007). El detalle sobre las metodologías en la literatura para las secuencias no AMPs recuperadas se muestran en la Tabla 16.

Para obtener un súper conjunto no redundante de posibles secuencias no AMP, analizamos las secuencias por longitud y por nivel de superposición entre los conjuntos de datos. Para diez de los once conjuntos de datos, la mayoría de sus secuencias peptídicas se encuentran en el intervalo de 5 a 100 residuos (ver Figura 23). Además, sólo tres conjuntos de datos comparados por pares tienen secuencias duplicadas (ver Figura 24); NAMP_ANFIS (Fernandes *et al.*, 2012) y NAMP_IAMP13 (Xiao *et al.*, 2013) tienen nueve secuencias idénticas, NAMP_CAMP14 (Waghu *et al.*, 2014) y NAMP_IAMP13 (Xiao *et al.*, 2013) tienen dos secuencias idénticas; NBactibase_DAMPD (Gabere y No-

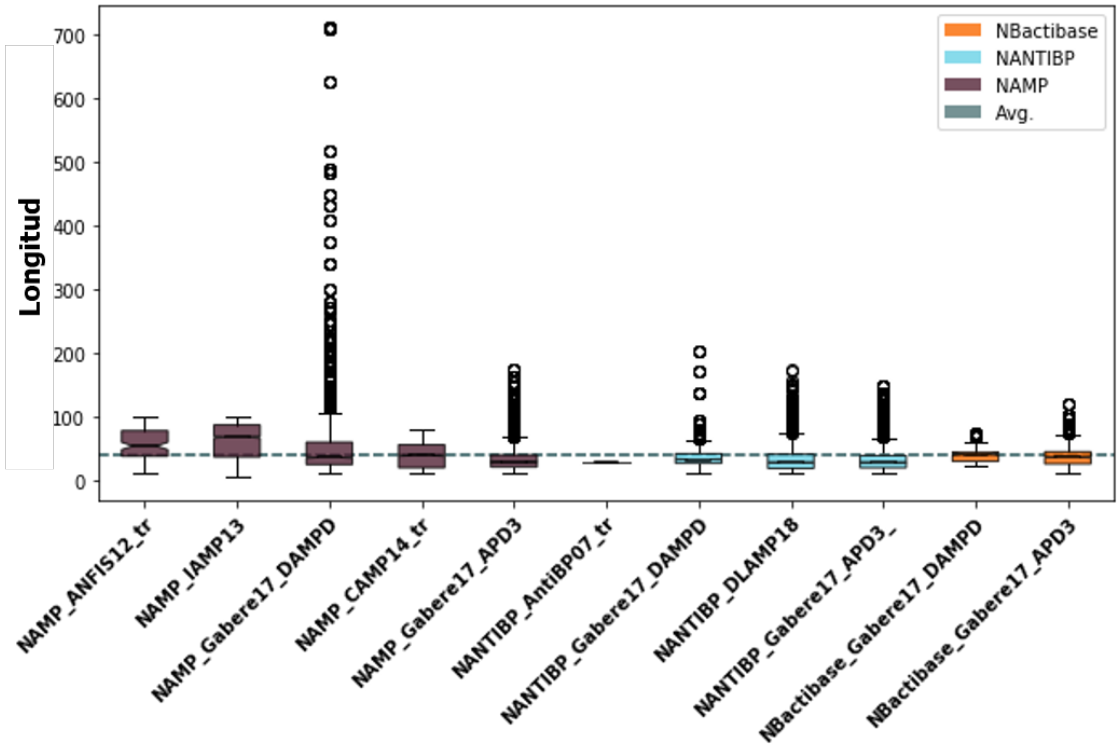


Figura 23. Distribución de la longitud de las secuencias de péptidos no AMP para once conjuntos de datos.

ble, 2017) y NANTIBP_DAMPD tienen 155 secuencias idénticas (Gabere y Noble, 2017). Como resultado de este análisis eliminamos las secuencias que tienen una longitud superior a los 100 aminoácidos, como también, secuencias duplicadas. En esta etapa recogimos 28748 secuencias potencialmente sin la actividad antimicrobiana.

Por otra parte, si bien las secuencias recuperadas no tienen anotación alguna de *Gene Ontology* (GO) relacionada con la respuesta de defensa hacia los microbios (ver Tabla 16), esto no es garantía de la ausencia de la actividad. Por esta razón, realizamos un alineamiento local por pares entre 28748 posibles no AMPs contra las 29400 AMPs verificados experimentalmente, esto con el fin de eliminar las secuencias homólogas. Utilizamos BLAST 2.6 para la alineación de secuencias, en donde se utilizaron las secuencias de AMPs como base de datos y las no AMPs como secuencias consulta. La Figura 25 muestra 4586 alineaciones locales por pares con un valor *e* menor a 1 entre conjuntos de datos AMPs y no AMPs. La similitud para estas coincidencias tiene una identidad de secuencia porcentual en el intervalo entre 20 y 100, donde el número de secuencias idénticas entre ambas bases de datos fue de 48. La mayoría de estas secuencias son patentes que no tienen la anotación de GO relacionada con la actividad

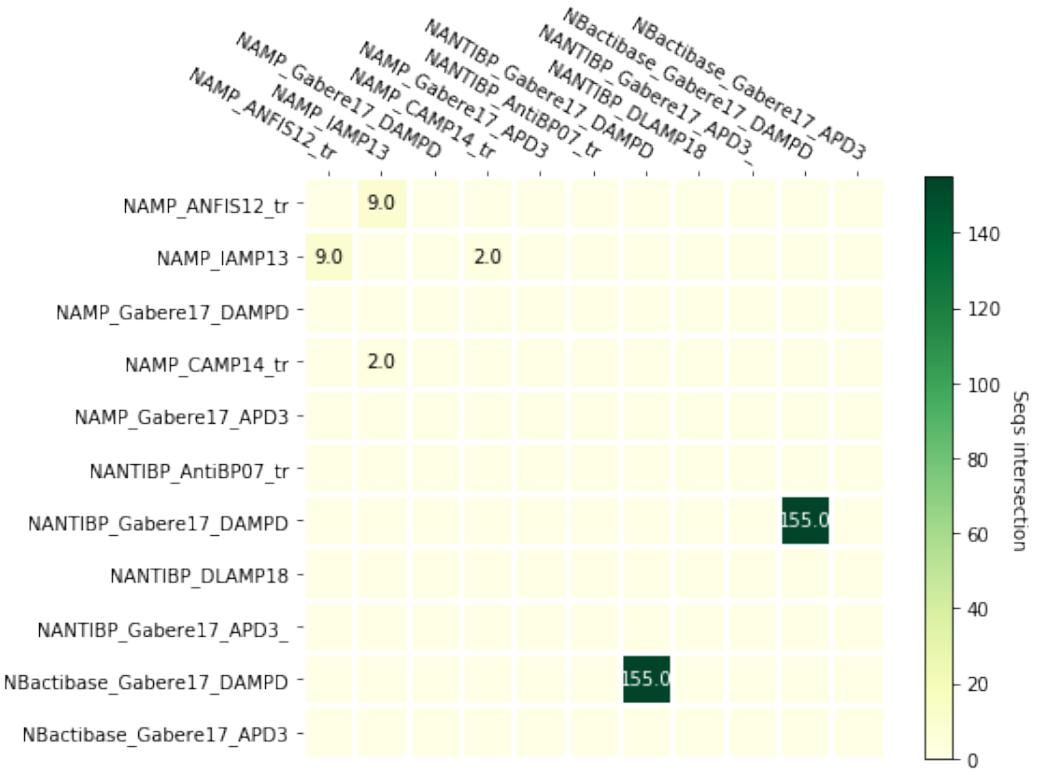


Figura 24. Intersección de las secuencias entre conjuntos de datos no antimicrobianos.

antimicrobiana en la base de datos de Uniprot.

Adicionalmente, utilizamos el predictor AMP accesible desde la web, CAMPR3, para predecir secuencias en ambos conjuntos de datos, es decir, los AMPs validados experimentalmente y los potenciales no AMPs. La Figura 26 muestra el resultado de la predicción antimicrobiana utilizando cinco modelos de clasificación diferentes: máquina de soporte vectorial (SVM), bosques aleatorios (RF), análisis de discriminantes (DA), red neuronal artificial (ANN) y el sistema de votación de las salidas de los tres modelos (*i.e.,* SVM, RF, DA). Los resultados muestran que CAMPR3 sólo puede clasificar correctamente el 67 % de las secuencias AMP. Por otro lado, el 15 % de los no AMPs se clasifican como AMPs.

Los resultados mostrados arriba llevan a la pregunta natural de qué secuencias se deben incluir como parte del conjunto no antimicrobiano. Para descartar cualquier falso negativo en el conjunto de no AMPs, hemos eliminado las secuencias que tienen al menos un 40 % de identidad y las secuencias identificadas como antimicrobianas por el clasificador CAMPR3 de votación. Finalmente, el conjunto no AMP resulta en 22688 péptidos no antimicrobianos potenciales en un intervalo entre 5 y 100 aminoácidos.

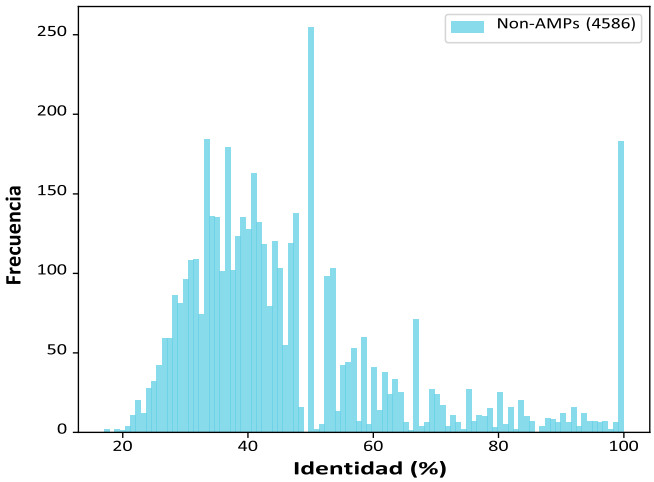


Figura 25. Distribución del porcentaje de identidad de las secuencias entre conjunto de datos de no AMPs y el conjunto de datos de AMPs. Se utilizó un valor de corte de *e-value* de 1 para seleccionar los alineamientos. El cálculo se realizó con BLAST.

6.1.2. Representación de los péptidos

Utilizamos dos tipos de representación de los péptidos; la primera es una codificación en caliente (*one-hot*), la cual se utiliza para representar las secuencias peptídicas en el modelo generativo; la segunda es un vector de características para el modelo de predicción de la actividad antimicrobiana.

6.1.2.1. Representación *one-hot*

Para convertir las secuencias peptídicas en una codificación en caliente (*one-hot*), seguimos una metodología similar a la propuesta en el trabajo de Müller *et al.* (2018). En esta metodología, primero a cada secuencia se añade el símbolo 'B' en el N-terminal de la misma (ver Figura 27.1). Enseguida, se identifica la secuencia más larga del conjunto de datos y el resto de las secuencias se rellenan con el símbolo de espacio ' ' hasta completar la longitud más larga (ver Figura 27.2). A continuación, para generar los ejemplos del conjunto de entrenamiento, para cada secuencia $s = \langle s_1 s_2 \dots s_n \rangle$ en el conjunto de datos AMP usamos la subcadena $\langle s_1 s_2 \dots s_{n-1} \rangle$ como una instancia y la subcadena $\langle s_1 s_2 \dots s_n \rangle$ como la etiqueta de salida, respectivamente (*i.e.,* el prefijo y el sufijo de s de longitud $n-1$) (ver Figura 27.3). Por último, para una correcta

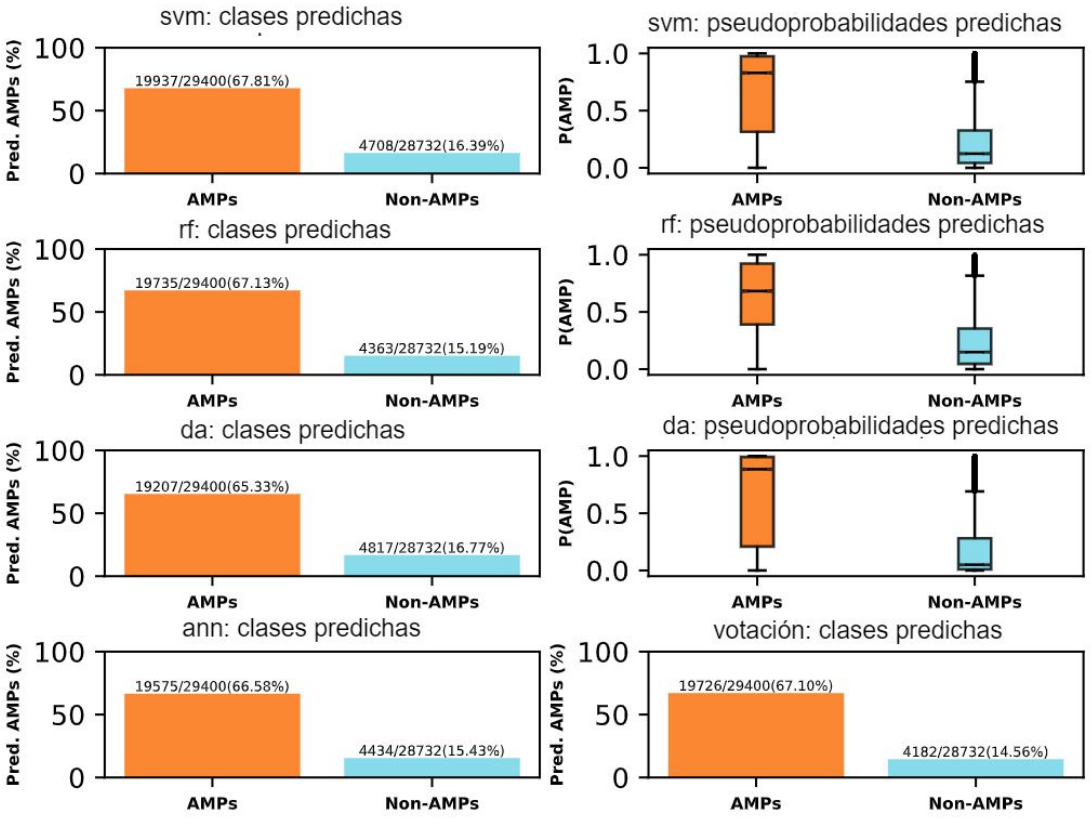


Figura 26. Resumen de los resultados de la predicción de CAMPR3 sobre los conjuntos de datos AMP y no AMP. Los algoritmos de aprendizaje de máquina son: máquina de soporte vectorial (SVM), bosque aleatorio (RF), análisis de discriminantes (DA), red neuronal artificial (ANN) y el sistema de votación resultado de los tres modelos (*i.e.*, SVM, RF, DA).

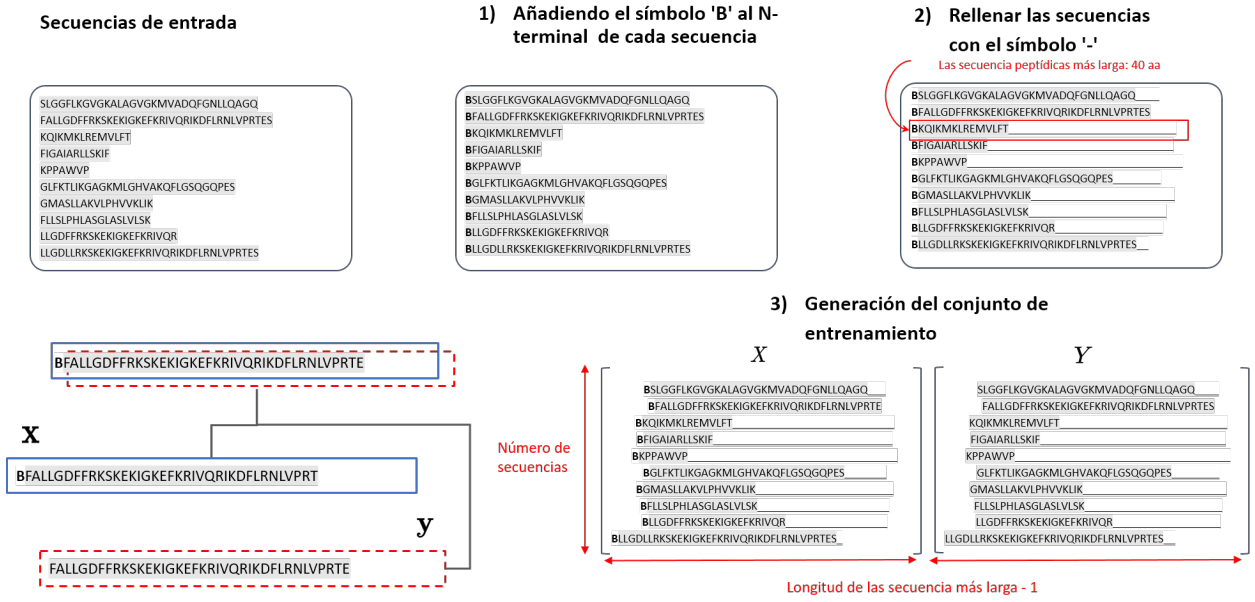


Figura 27. Esquema de la metodología para la generación del conjunto de entrenamiento para el modelo generativo.

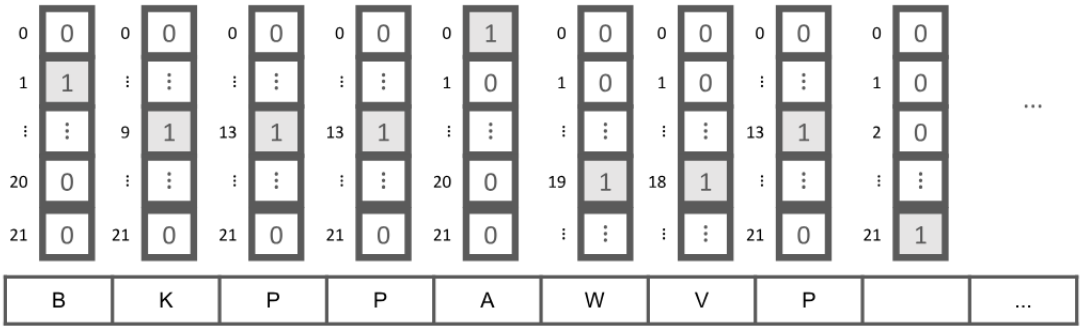


Figura 28. Ejemplo de una secuencia de péptido representada en una codificación *one-hot*. El cuadrado gris con un valor es el índice representado para un símbolo particular.

representación en el modelo generativo, transformamos cada aminoácido en las secuencias a una codificación *one-hot*. La codificación *one-hot* consiste en transformar cada aminoácido en un vector binario de longitud igual al vocabulario Σ. Σ es un vector compuesto por los 20 aminoácidos estándar, representados en código de una sola letra, el símbolo *B* y el espacio (`´), este se puede enumerar de la siguiente manera:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Σ = [	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	]

Por lo tanto, podemos representar cada símbolo en Σ como un vector binario de longitud 22, donde todas las posiciones son cero excepto la posición igual al índice del símbolo en el vocabulario que toma uno.

Un ejemplo de una secuencia representada en una codificación *one-hot* se muestra en la Figura 28, donde el cuadrado gris con valor de uno indica la representación del índice para un símbolo en particular. Por lo tanto, para una secuencia de tamaño n, su representación en una codificación *one-hot*, es una matriz de $n \times |\Sigma|$. Por lo tanto, el conjunto de datos de entrenamiento, compuesto por m secuencias, para el modelo generativo $\mathcal{D}_{gen}$ está compuesto por el tensor $\mathbf{X}$ y $\mathbf{Y}$ de tamaño $m \times n \times |\Sigma|$, donde $\mathbf{X}_{i,:}$ es la instancia de entrada $\mathbf{x}^{(i)}$ del conjunto de datos y $\mathbf{Y}_{i,:}$ es la etiqueta objetivo $\mathbf{y}^{(i)}$ asociada a la instancia $\mathbf{x}^{(i)}$, respectivamente (ver Figura 29).

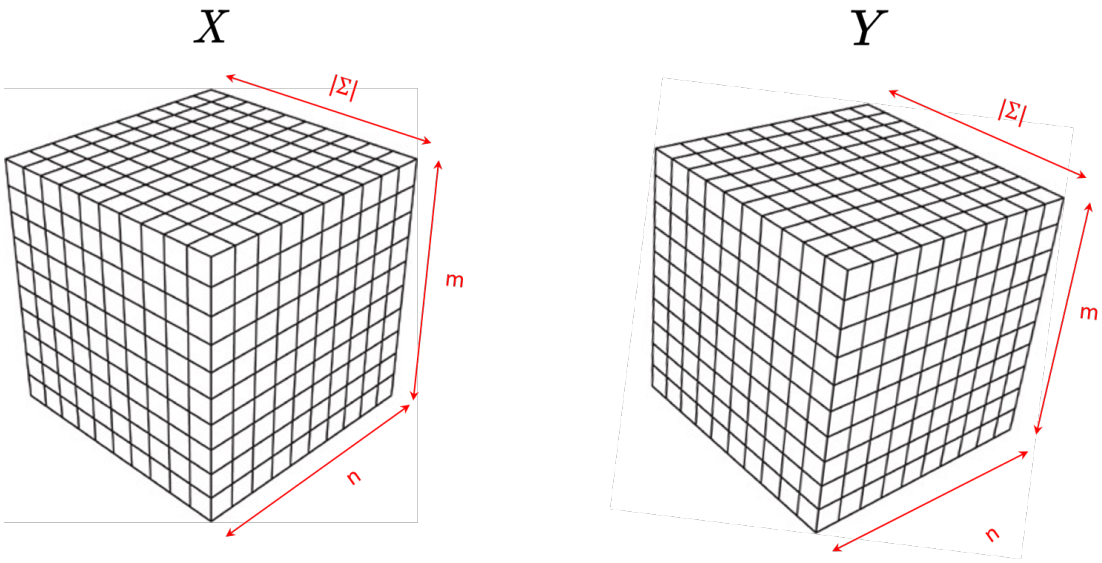


Figura 29. Esquema del conjunto de entrenamiento para el modelo generativo. *X* representa el conjunto de instancias, mientras que *Y* es el conjunto de etiquetas para las instancias. *X* e *Y* están compuestos de *m* secuencias de *n* caracteres cada uno, donde cada carácter está representado por un vector del tamaño |Σ|.

6.1.2.2. Representación basada en descriptores moleculares

Para el modelo de predicción de la actividad antimicrobiana utilizamos una representación basada en descriptores moleculares. Para convertir las secuencias en vectores de características, se utilizaron los descriptores moleculares, MODAMP (Beltrán *et al.*, 2017; Beltran *et al.*, 2018) (Ver Capítulo 3). MODAMP calcula un total de 122 descriptores moleculares de los cuales se genera un vector de 268 dimensiones para cada secuencia. El número de descriptores moleculares para cada propiedad fue: 74 en conformación, 10 de carga, 31 de carácter hidrofóbico y 2 en otras propiedades. Por lo tanto, el conjunto de datos de entrenamiento para el modelo de predicción $\mathcal{D}_{pred}$ está compuesto por $m \times (268 + 1)$ cuyas filas están dadas por el ejemplo de entrada $(\mathbf{x}^{(i)})^T$ y y_i es el objetivo asociado a $(\mathbf{x}^{(i)})^T$, *i.e.*, denotan que el péptido tiene o no la actividad antimicrobiana.

6.1.3. Modelo generativo

El problema para diseñar un nuevo péptido antimicrobiano puede ser modelado como un patrón de secuencias en el que dado el símbolo de inicio 'B', el modelo genere

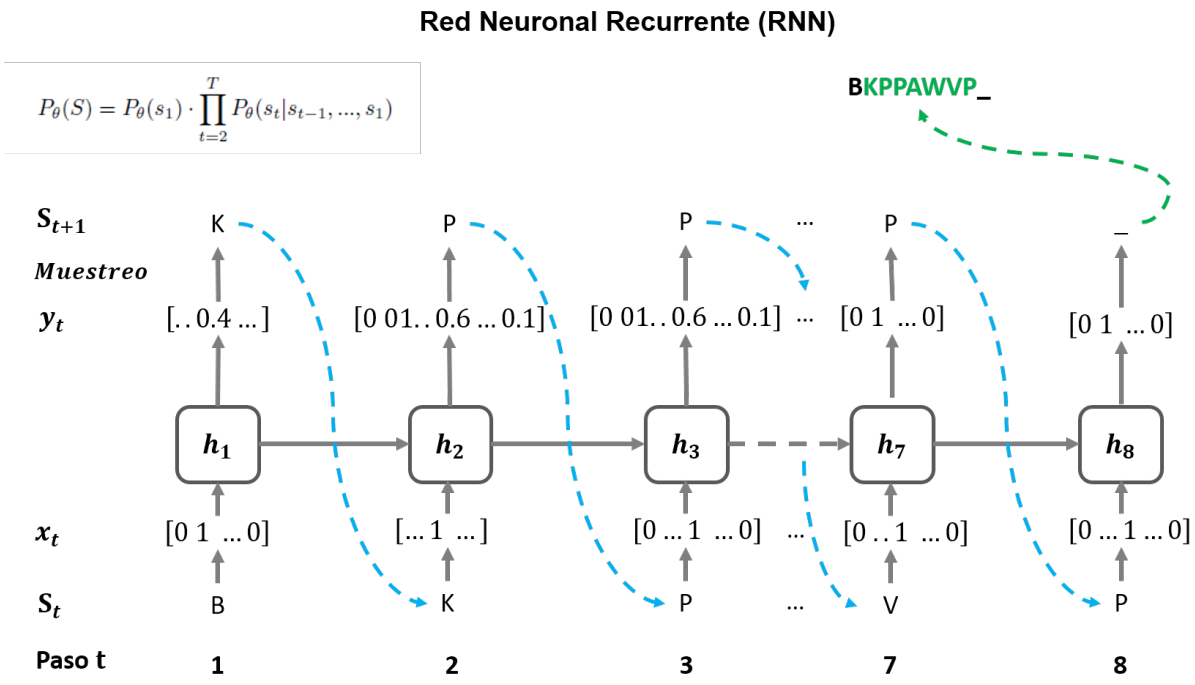


Figura 30. La generación de símbolos y el proceso de muestreo para la generación de una nueva secuencia de péptido.

una secuencia de aminoácidos proveniente de la distribución estimada del espacio de secuencias antimicrobianas (ver Figura 30). A este tipo de modelos se le denomina red neuronal recurrentes (RNN) de uno a muchos. En contraste con una red neuronal *feedforward* que aprende las reglas de manera independiente por cada residuo de las secuencias de péptidos, las RNNs son capaces de usar el contexto anterior a través de compartir sus estados ocultos anteriores.

En este estudio se utilizó una red neuronal profunda bidireccional de memoria a largo y corto plazo para generar nuevos péptidos con potencial actividad antimicrobiana. Los RNN bidireccionales (BRNN) son una extensión de los RNN convencionales a partir de los cuales se pueden procesar las secuencias peptídicas en ambas direcciones, conectando dos capas ocultas de direcciones opuestas a la misma salida, es decir, las secuencias ocultas hacia adelante y las secuencias ocultas hacia atrás. Por otro lado, el término profundo se debe a que creamos nuestro modelo apilando múltiples BRNN. Formalmente, definimos nuestro BRNN de la siguiente manera.

Dado una secuencia de entrada $\mathbf{x} = (x_1, \ldots, x_T)$ (*i.e.,* representadas en una codificación *one-hot*), la BRNN calcula el vector de secuencias ocultas hacia adelante y hacia atrás $\overrightarrow{\mathbf{h}}^l$, $\overleftarrow{\mathbf{h}}^l$, y el vector de secuencias de salida $\hat{\mathbf{y}} = (\hat{y_1}, \ldots, \hat{y_T})$ donde $\hat{y_t}$ es el

vector de salida con una distribución de probabilidad sobre el vocabulario Σ del siguiente residuo x_{t+1} en la secuencia. La capa oculta se calcula iterando por nivel de espacio (es decir, las capas ocultas) de $l = 1$ a L y por el nivel de tiempo en ambas direcciones, es decir, la capa hacia adelante de $t = 1$ a T y la capa hacia atrás de $t = T$ a 1 (ver Figura 31):

$$\overrightarrow{h}_t^l = \mathfrak{g}(W_{\overrightarrow{h^l}\overrightarrow{h^l}} \overrightarrow{h}_{t-1}^l + W_{\overrightarrow{h^{l-1}}\overrightarrow{h^l}} \overrightarrow{h}_t^{l-1} + W_{\overleftarrow{h^{l-1}}\overrightarrow{h^l}} \overleftarrow{h}_t^{l-1} + b^l), \tag{39}$$

$$\overleftarrow{h}_t^l = \mathfrak{g}(W_{\overleftarrow{h^l}\overleftarrow{h^l}} \overleftarrow{h}_{t+1}^l + W_{\overrightarrow{h^{l-1}}\overrightarrow{h^l}} \overrightarrow{h}_t^{l-1} + W_{\overleftarrow{h^{l-1}}\overrightarrow{h^l}} \overleftarrow{h}_t^{l-1} + b^l) \tag{40}$$

y

$$\hat{y_t} = \phi(W_{\overrightarrow{h^L}y} \overrightarrow{h}_t^L + W_{\overleftarrow{h^L}y} \overleftarrow{h}_t^L + b^y), \tag{41}$$

donde $\mathfrak{g}$ es la función de la capa oculta, W denota las matrices de peso, en donde los subíndices indican la fuente y el destino (*e.g.*, $W_{\overleftarrow{h^l}\overleftarrow{h^l}}$ es la matriz desde la capa oculta hacia delante l para la capa oculta hacia delante l), b denota el vector de sesgo. Es importante notar que $\mathbf{h^0} = \mathbf{x}$. Ademas, ϕ es la función de activación *softmax*. Para calcular $\mathfrak{g}$, usamos la memoria a largo y corto plazo (LSTM por sus siglas en inglés de *Long Short-Term Memory*) introducida por Hochreiter y Schmidhuber (1997).

La Figura 31 ilustra nuestro modelo LSTM RNN bidireccional profundo, donde el rectángulo redondeado representa una capa de LSTM con 256 unidades de memoria, y la flecha representa matrices de conexión a diferentes niveles, es decir, el tiempo y el espacio. Como salida, tenemos una capa que contiene 22 neuronas con una función de activación *softmax* ϕ. El LSTM RNN bidireccional profundo fue optimizado con Adam (Kingma y Ba, 2014) utilizando lotes de 128 instancias, un número de épocas de 20 y una tasa de aprendizaje de 0.0008. Adicionalmente, utilizamos la función de pérdida categórica de entropía cruzada (*categorical cross-entropy*) (Goodfellow *et al.*, 2016).

6.1.3.1. Generación de secuencias

Para el muestreo de las secuencias utilizamos el modelo generado en la Subsección 6.1.3 para producir una serie de secuencias. En esta etapa, indicamos como entrada: el

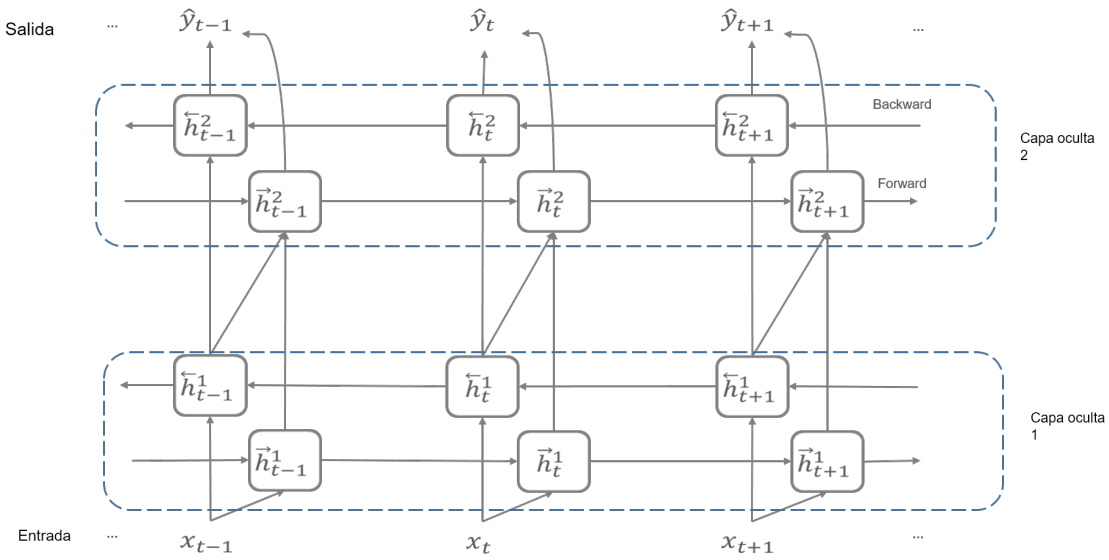


Figura 31. Red neuronal bidireccional recurrente de memoria a largo y corto plazo. Esquema de la arquitectura aplicada para generar potenciales secuencias de péptidos antimicrobianos. El rectángulo redondeado representa una capa de memoria a largo y corto plazo, y la flecha horizontal representa las matrices de conexión en el momento del paso, la flecha derecha representa las secuencias ocultas hacia adelante y la flecha izquierda las secuencias ocultas hacia atrás. La flecha vertical representa las matrices de conexión a nivel espacial.

número de muestras, la longitud mínima de secuencias y la temperatura. Un esquema de este proceso se muestra en la Figura 32.

Para producir una secuencia peptídica, comenzamos con el símbolo $s_1 = B$ convertido en su vector *one-hot* correspondiente y se da como primera entrada $x^{<1>}$ al modelo generativo. El modelo produce y_1, que es la distribución de probabilidad sobre el vocabulario Σ para obtener el siguiente residuo s_2 de la secuencia de péptido, tal como se indica en Müller *et al.* (2018). Un símbolo s se muestrea de la distribución de probabilidad y_1. Si s es uno de los 20 aminoácidos entonces será un residuo de las secuencias $< s_1, s_2 >$ y se dará como entrada del modelo $x^{<2>}$, de lo contrario, la generación de la secuencia se termina (ver ejemplo de la Figura 30).

6.1.4. Predicción de la actividad antimicrobiana

Para verificar si las secuencias generadas son antimicrobianas implementamos una red neural profunda (DNN). La DNN está entrenada utilizando péptidos antimicrobianos validados experimentalmente y péptidos potenciales no antimicrobianos representados en el espacio de los descriptores moleculares. La DNN entrenada recibe como entrada un nuevo péptido generado, representado en el espacio de los descriptores moleculares, y da como salida la pseudoprobabilidad del péptido de tener la actividad

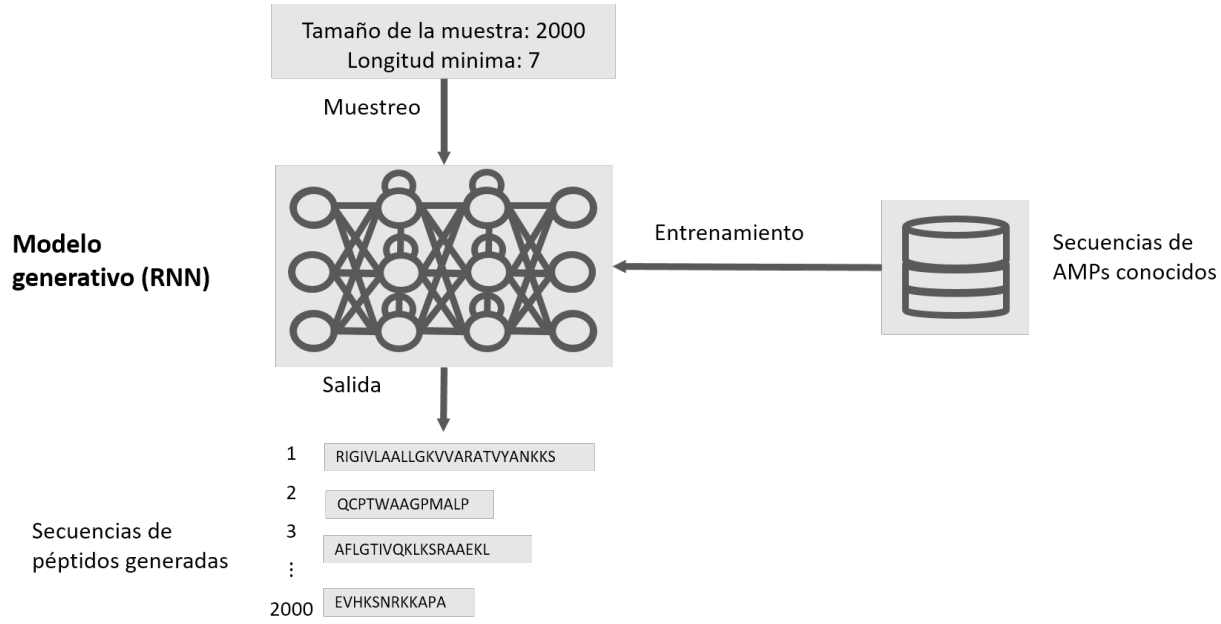


Figura 32. Esquema general del modelo generativo.

antimicrobiana y la etiqueta de la clase (*i.e.,* AMP o no AMP). En la Figura 33 se presenta el esquema general para la predicción de la actividad de una nueva secuencia.

A continuación, describimos el modelo de línea base para comparar el rendimiento y la complejidad con nuestra propuesta de DNN. También describimos como obtener de forma eficiente la arquitectura óptima de la red neuronal profunda y los hiperparámetros adecuados.

6.1.4.1. Modelo línea base: red neuronal simple

Se utilizó una red neural simple (SNN) como modelo de referencia (línea base). SNN tiene como capa de entrada un vector m-dimensional, $\left[x_1^i, \ldots, x_m^i\right]^T$, y como salida una capa logística que genera la probabilidad del péptido a ser un AMP. En este trabajo consideramos dos modelos SNN: (1) SNN (todas las características) en donde cada péptido está representado con 268 descriptores moleculares; (2) SNN (funciones de filtro) en donde cada péptido está representado con 243 descriptores moleculares. La razón por la que elegimos la SNN es porque es la red neuronal más simple, *i.e.,* tiene sólo una capa con una unidad como salida.

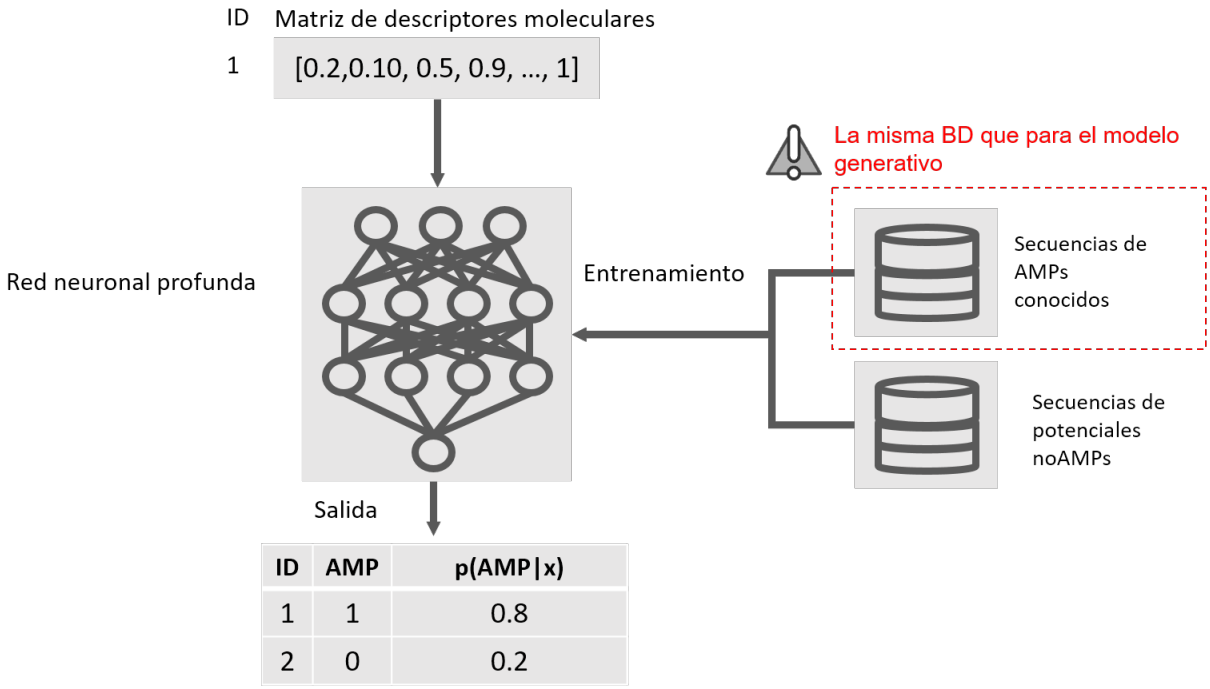


Figura 33. Esquema general del modelo clasificación.

6.1.4.2. Modelo red neural profunda y ajuste de hiperparámetros

Para obtener los hiperparámetros adecuados y la mejor arquitectura de nuestra red neuronal profunda *feedforward* (DFN), empleamos Sherpa (Hertel *et al.*, 2018). Sherpa es una librería gratuita de código abierto de optimización de hiperparámetros para modelos de aprendizaje de máquina. En sherpa, se seleccionó una optimización bayesiana para explorar eficientemente el espacio hiperparamétrico. Para estimar la calidad de un conjunto de hiperparámetros seleccionados, se utilizó la validación cruzada quíntuple de la función binaria de costo de entropía cruzada *cross-entropy* (Goodfellow *et al.*, 2016).

Sintonizamos los siguientes hiperparámetros: número de capas ocultas, número de unidades ocultas por capa, número de épocas, probabilidad de interrupción (*dropout*), término de penalización L2, optimizador, método de inicialización de pesos, función de activación, la restricción de peso, tasa de aprendizaje y el decaimiento de pesos (ver Tabla 17).

Finalmente, la capa de salida en la DFN contiene una sola neurona y utiliza una función de activación sigmoide para producir la salida de pseudoprobabilidad de que

Tabla 17. Los hiperparámetros considerados en la optimización de la red neuronal profunda para el problema de la clasificación binaria de la actividad antimicrobiana.

Nombre	Tipo	Intervalos
Entrenamiento		
Tamaño del lote	Discreto	[1,300]
Época	Ordinal	[10, 20, 50, 100, 200]
Optimizador		
Algoritmo de optimización	Opción	['Adadelta', 'Adagrad', 'Adam', 'Adamax', 'Nadam']
Tasa de aprendizaje	Continuo	[0.001,1]
Decaimiento	Continuo	[0.0, 0.2]
Arquitectura		
Capas ocultas	Discreto	[0, 5]
Neuronas	Discreto	[67, 250]
Función de activación	Opción	['elu', 'selu','relu', 'tanh', 'softmax', 'softplus', 'softsign', 'sigmoid', 'hard_sigmoid', 'linear', 'exponential']
Regularización		
Regularización de parámetros	Opción	[None, L2]
Tasa de regularización	Ordinal	[0, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10]
Probabilidades de interrupción	Ordinal	[0.1, 0.2, 0.3, 0.4]
Otros		
Inicialización del kernel	Opción	['uniform', 'lecun_uniform', 'normal', 'glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']
Restricción de peso	Ordinal	[None,1, 2, 3, 4, 5]

la entrada sea antimicrobiana.

6.1.5. Filtrado de secuencias de péptidos que superan un umbral de similitud

Realizamos un alineamiento local por pares entre las secuencias de nuestros péptidos diseñados y el conjunto de datos AMP (ver subsección 6.1.1.1) para determinar el grado de similitud entre las nuevas secuencias y las secuencias de AMPs conocidas. Seleccionamos los alineamientos locales por pares entre un péptido diseñado y la secuencia más cercana en el conjunto de AMPs, *i.e.,* el alineamiento con el porcentaje más alto de identidad. Para el alineamiento de las secuencias, utilizamos BLAST 2.6 en su versión *standalone*.

6.1.6. Selección de nuevos AMPs diversos

Para seleccionar un subconjunto no redundante de las secuencias generadas se propone generar conglomerados basados en el porcentaje de identidad. Después, seleccionar una secuencia representante por conglomerado. Lo anterior es con el objetivo de entregar diversidad a los evaluadores de las secuencias en laboratorio. La generación de secuencias diversas es significativa porque los péptidos candidatos pueden

fallar en etapas posteriores del proceso de descubrimiento de fármacos (Benhenda, 2017).

6.2. Resultados Preliminares

En esta sección se presenta los resultados preliminares de dos de los cuatro componentes del enfoque para el diseño *in silico* de AMPs. El primer componente es el modelo de predicción de la actividad antimicrobiana. El segundo componente es el modelo generativo de secuencias de péptido.

6.2.1. Predicción de la actividad antimicrobiana

A continuación, se presenta la mejor arquitectura encontrada por Sherpa (Hertel *et al.*, 2018) y el desempeño que se obtuvo tanto en el conjunto de entrenamiento como en el de pruebas.

6.2.1.1. Arquitectura de los modelos y entrenamiento

La mejor arquitectura para el DFN encontrada por Sherpa (Hertel *et al.*, 2018) fue una red neuronal de 4 capas, el esquema de la arquitectura se ilustra en la Figura 34. El DFN incluye una capa de entrada con 268 y 243 características para el DFN (todas las características) y DFN (características de filtro), respectivamente; tres capas ocultas que contienen 201 neuronas con la función de activación *ReLU*, la probabilidad de interrupción fue de $(0.3)h_i$, donde h_i es la posición de la capa oculta. Como salida, una capa con una neurona con función sigmoide. Para los pesos iniciales de las capas se utilizó una inicialización uniforme aleatoria. El DFN fue optimizado con Adadelta (Zeiler, 2012) con un tamaño de lote de 293 instancias, una tasa de aprendizaje de 0.666, un decaimiento de la tasa de aprendizaje por época de $1.82e-7$, un número de época de 20, la regularización de L2 a 0.0, y las limitaciones de peso de la norma máxima igual a 3.

El mejor modelo DFN con todas las características y filtradas obtuvo una función de costo promedio en la validación de 0.218 y 0.219, respectivamente. El desempeño de

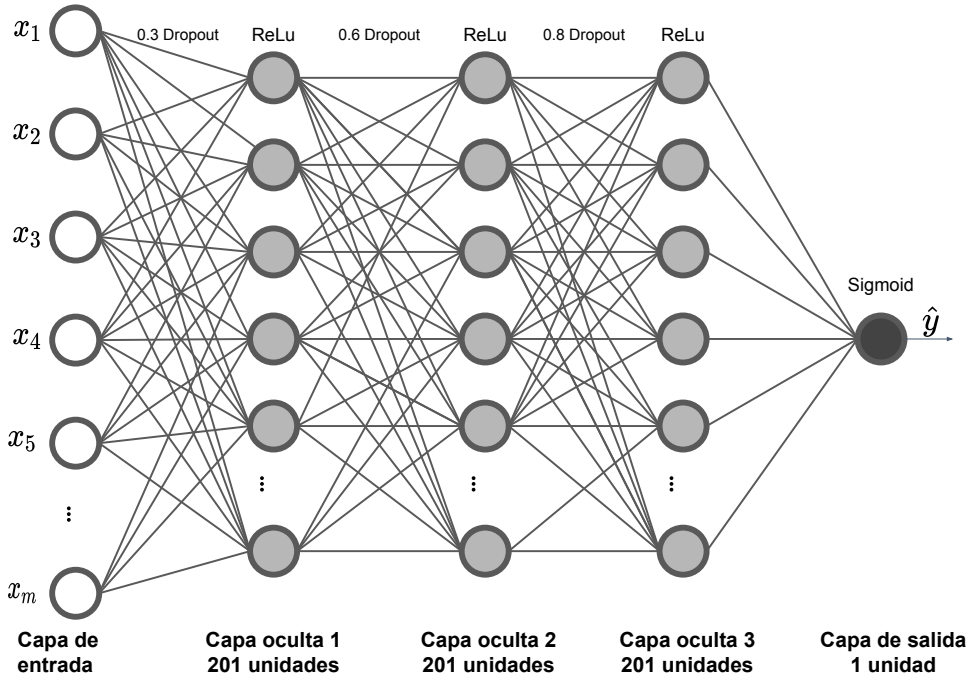


Figura 34. Esquema de la arquitectura de la red neural profunda e hiperparámetros óptimos encontrados por Sherpa (Hertel *et al.,* 2018) para la clasificación binaria de la actividad antimicrobiana. La capa de entrada (blanca) es un vector *m*-dimensional que captura las propiedades fisicoquímicas en valores reales, cada componente codifica el valor de un descriptor molecular en particular. Cada capa oculta (gris) de 201 unidades con la función de activación ReLU. Se aplicó *dropout* a las capas ocultas con una probabilidad de (0.3) × número de capa oculta. La capa de salida (gris oscuro) es una regresión logística que produce una probabilidad de ser antimicrobiana entre 0 y 1.

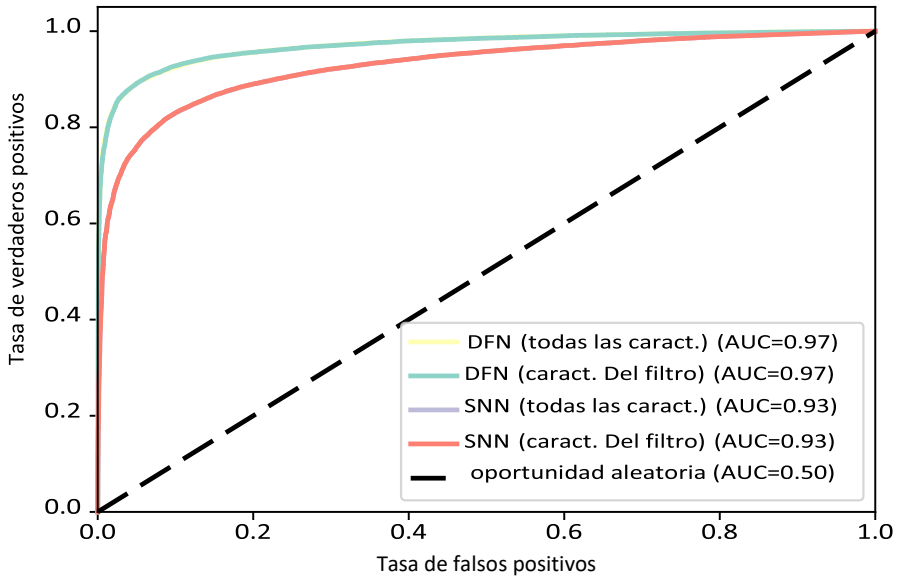


Figura 35. Curva ROC para la clasificación binaria de AMPs en la validación cruzada de 5 pliegues. Se comparan la mejor arquitectura para la red neuronal profunda (DFN), el modelo de línea de base (SNN), y un clasificador aleatorio. Los valores al lado de cada clasificador indican el área bajo la curva ROC (AUC), un mayor valor de AUC significa un mejor clasificador.

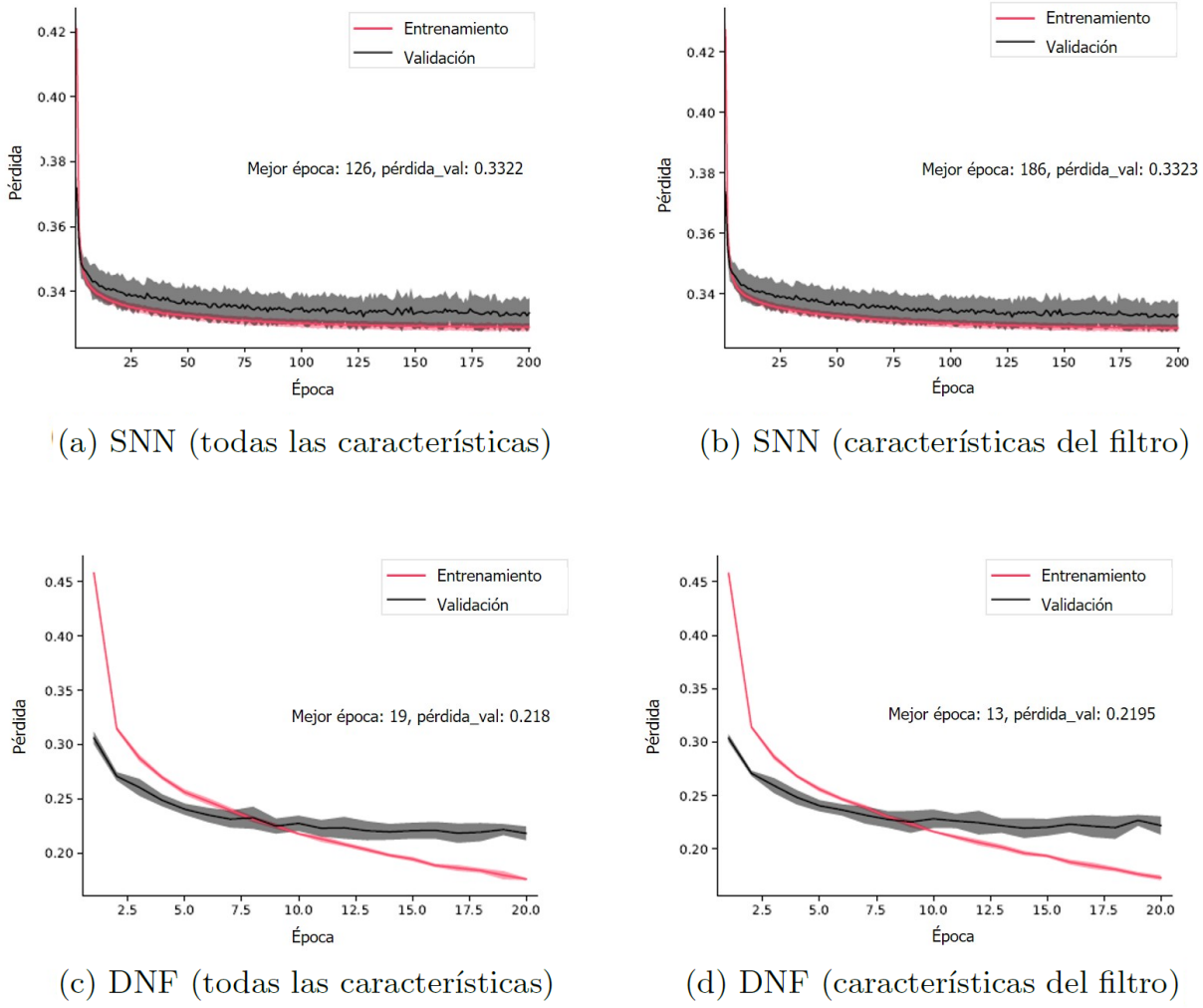
Tabla 18. Desempeño de la validación cruzada de 5 pliegues para el problema de predicción de la actividad antimicrobiana.

Método	Sens(%)	Spec(%)	Prec(%)	Acc(%)	MCC	AUC
SNN (todas las características)	85.76 (0.67)	86.31 (1.05)	88.97 (0.60)	86.01 (0.15)	0.72 (0.00)	0.93 (0.00)
SNN (características del filtro)	85.73 (0.59)	86.32 (1.02)	88.97 (0.60)	85.99 (0.16)	0.72 (0.00)	0.93 (0.00)
DFN (todas las características)	91.09 (0.83)	92.39 (0.89)	93.91 (0.58)	91.66 (0.19)	**0.83** (0.00)	**0.97** (0.00)
DFN (características del filtro)	**91.25** (0.79)	**92.50**(0.80)	**94.00** (0.51)	**91.80** (0.12)	**0.83** (0.00)	**0.97** (0.00)

la validación cruzada y entrenamiento de nuestro modelo DFN para la función binaria de la entropía cruzada (*cross-entropy*) por época se muestra en la Figura 36. El DFN (todas las características) y DFN (características del filtro) lograron un AUC de 0.97 (ver Figura 35), lo que significa que los modelos generados tienen una excelente capacidad para aprender qué es un péptido antimicrobiano. También encontramos que el DFN supera al mejor modelo de línea base para todas las métricas (ver Tabla 18 para más detalles).

6.2.2. Evaluación del modelo

Medimos la capacidad de predicción del DFN sobre nuevas secuencias de péptidos, es decir, secuencias peptídicas que no han sido utilizadas en el proceso de ajuste de hiperparámetros para elegir la mejor arquitectura para la DFN (ver Sub-



(a) SNN (todas las características) (b) SNN (características del filtro)

(c) DNF (todas las características) (d) DNF (características del filtro)

Figura 36. El promedio y la desviación estándar de los parámetros críticos. Función de costo dentro de una validación cruzada de 5 pliegues (un modelo con una menor pérdida de costo es mejor). El promedio de la función de costo (línea sólida) y la desviación estándar (áreas sombreadas) para el conjunto de entrenamiento y la validación. (a)-(b) Comportamiento de la red neuronal simple (SNN) utilizada como modelo de línea base, (a) SNN con capa de entrada de todas las características (268 descriptores moleculares) y (b) SNN con capa de entrada de sólo las características del filtro (243 descriptores moleculares). (c)-(d) Rendimiento de la Red de Retroalimentación Profunda (DFN) afinada.

Tabla 19. Comparación de rendimiento en la partición de pruebas del conjuntos de AMPs.

Method	Sens(%)	Spec(%)	Prec(%)	Acc(%)	MCC	AUC
CAMPR3-SVM	67.56	95.86	95.65	79.62	0.639	0.849
CAMPR3-RF	67.16	96.24	96.01	79.55	0.640	0.896
CAMPR3-ANN	66.26	92.27	92.03	77.34	0.588	0.793
CAMPR3-DA	65.23	95.38	95.00	78.08	0.613	0.854
CAMPR3-Voting	66.96	**100.00**	**100.00**	81.04	0.681	0.835
SNN (todas las características)	85.99	87.10	89.98	86.46	0.726	0.931
SNN (características del filtro)	85.95	87.10	89.98	86.44	0.726	0.931
DFN (todas las características)	**92.95**	91.32	93.52	92.26	0.842	**0.974**
DFN (características del filtro)	92.59	92.30	94.18	**92.47**	**0.847**	**0.974**

sección 6.1.4.2). Comparamos nuestro modelo DFN utilizando la partición de prueba del conjunto de datos (ver Subsección 6.1.1) con cuatro métodos de aprendizaje de máquina de última generación para el reconocimiento de AMPs (Waghu *et al.*, 2014). CAMPR3 (Waghu *et al.*, 2014) tiene cuatro predictores AMP accesibles a través de la web (http://www.camp.bicnirrh.res.in/predict/) estos son: un clasificador creado por una máquina soporte vectorial con kernel polinomial (grado 4), CAMPR3-SVM; un bosque aleatorio, CAMPR3-RF; un clasificador creado mediante el uso de una red neural artificial (CAMPR3-ANN); y análisis de discriminantes (CAMPR3-DA).

En la Tabla 19 el resultado en negrita indica el mejor rendimiento para un criterio determinado. Nuestro modelo DFN supera a CAMPR3 (Waghu *et al.*, 2014) en las siguientes medidas: Sens(%) con 92.59, Acc(%) con 92.47, MCC con 0.847, y AUC con 0.974. También, encontramos que nuestro DFN supera el modelo de votación de CAMPR. El modelo de votación CAMPR es un modelo que selecciona la etiqueta de predicción más frecuente de los modelos CAMPR3-SVM, CAMPR3-RF y CAMPR3-DA.

6.2.3. Evaluación de las secuencias generadas

A continuación, mostramos el desempeño que obtuvo nuestro modelo generativo bidireccional LSTM en el entrenamiento y la validación. Adicionalmente comparamos el desempeño del modelo bidireccional LSTM con el modelo de una sola dirección LSTM, con el objetivo de verificar si la función de costo disminuye cuando las secuencias se procesan en ambas direcciones. Los resultados indican que se obtiene un menor error utilizando el modelo bidireccional LSTM (ver Figura 40), en donde el error en la validación es de 0.0004 mientras que para el modelo LSTM es de 0.8309, respectivamente. En este sentido, es importante señalar que para ambos modelos (*i.e.,* LSTM y bidireccional LSTM) el costo de la evaluación es menor que el del entrenamiento, por lo

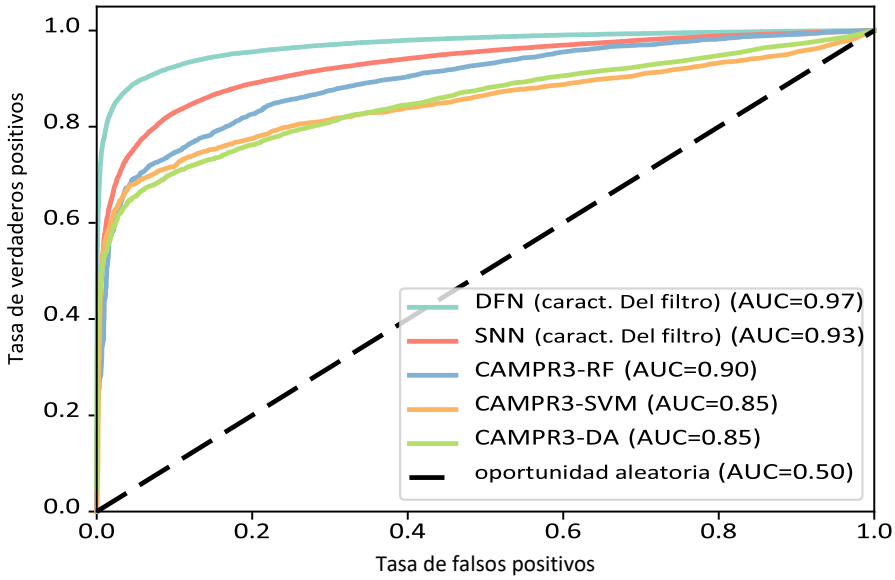


Figura 37. Curva ROC para la clasificación binaria de AMPs en el conjunto de datos de prueba. Se compara la mejor arquitectura para la red neuronal profunda *feedforward* (DFN) y el predictor AMP accesible a través de la web (CAMPR3). El DFN supera a CAMPR3 en la métrica de las AUC.

Tabla 20. Ejemplo de secuencias de péptidos generadas por el modelo generativo bidireccional LSTM entrenado.

ID	Secuencia	probabilidad
BiLSTM_2	AVRICTHICC	1
BiLSTM_16	CCAKKRRQQ	1
BiLSTM_20	AIKLKQAAYAH	1
BiLSTM_29	VRNWWPAV	1
BiLSTM_4	LLSKAAKEGA	0.98
BiLSTM_18	CTRFTLAK	0.99
BiLSTM_46	GKKFNQPHCYVLPCCH	1
BiLSTM_64	FTLVFMWEHTCSPVCC	0.89

que esto se puede deber al fenómeno conocido como *underfitting* (ver Apéndice A.3, Figura 40, línea sólida color negro).

6.3. Discusión

Nuestro enfoque pretende diseñar péptidos con actividad antimicrobiana potencial, generando los péptidos en el espacio de aminoácidos y evaluándolos en el espacio químico (*i.e.,* péptidos representados en descriptores moleculares). Con el objetivo de eliminar péptidos diseñados similares a AMPs conocidos utilizamos BLAST para su filtrado. Los resultados preliminares muestran un buen desempeño tanto en la validación

cruzada de 5 pliegues y con la partición de prueba. Lo anterior es importante señalarlo, debido a que la evaluación se realizó con un gran número de secuencias (52080 secuencias). Trabajos anteriores en la clasificación de AMPs llegan aproximadamente a las 8000 secuencias (Waghu *et al.*, 2014).

Por otro lado, dado que el modelo generativo y el modelo de clasificación se encuentran desconectados no es posible garantizar que todos los péptidos que se generen tendrán la actividad antimicrobiana potencial. Para este propósito, como trabajo futuro proponemos la unión de estos modelos utilizando las redes generativas adversarias.

Por último, relacionado con la generación de conjunto de datos negativos (*i.e.,* no AMPs), en este trabajo se obtiene un gran conjunto de datos con actividad no antimicrobiana potencial. En este sentido, queda evaluar la complejidad del conjunto negativo midiendo qué tan alejado se encuentra del conjunto positivo. Sin embargo, este problema de generar un conjunto de datos con péptidos sin la actividad antimicrobiana es todavía un problema por resolver.

Capítulo 7. Conclusiones y trabajo futuro

7.1. Conclusiones

En esta tesis se describe el proceso de diseño e implementación de algoritmos capaces de identificar péptidos antimicrobianos selectivos con una buena capacidad de discriminación apoyado por la evidencia biológica existente de péptidos antimicrobianos aislados y caracterizados de la naturaleza. A continuación se presenta un sumario del trabajo realizado seguido de las conclusiones de este trabajo de tesis (Ver resumen gráfico en las figuras 38 y 39). La descripción se hace en cuatro partes principales.

Primero, recolectamos e implementamos un conjunto de descriptores moleculares calculables en secuencias de péptidos, a la biblioteca le denominamos MODAMP (por sus siglas en inglés de MOlecular Descriptor for AntiMicrobial Peptides) (ver Capítulo 3). MODAMP tiene como objetivo ofrecer una alternativa libre y específica para péptidos (*i.e.,* no pretende sustituir los software existentes para el cálculo de descriptores). Los descriptores calculables en MODAMP están relacionadas a cinco propiedades principales: carga, conformación, carácter hidrofóbico, estructura secundaria y otras propiedades. Una versión de MODAMP se encuentra disponible en la base de datos StarPepDB (Aguilera-Mendoza *et al.*, 2019), en StarPepDB se pueden utilizar los descriptores moleculares para analizar las secuencias existentes de AMPs en el espacio químico. Si bien existen otras representaciones, tal como la estructura 3D que puede ofrecer mayor información acerca del péptido, en las bases de datos los péptidos se encuentran mayormente representados en su estructura primaria (*i.e.,* secuencia de aminoácidos), es por este motivo que decidimos contemplar hasta descriptores 1D en MODAMP.

Enseguida, teniendo una biblioteca para transformar del espacio de secuencias de aminoácidos al espacio químico, el siguiente propósito de este trabajo fue encontrar una reducción lineal del espacio químico de entrada, de tal manera que los péptidos con diferentes actividades biológicas tiendan a estar lejos unos de los otros, mientras que los AMPs tiendan a estar cercanos entre sí. Para encontrar este subespacio se modeló el problema como uno de optimización multiobjetivo de ponderación de los descriptores moleculares. Para resolver este problema, se utilizó una variante de una metodología general basada en un algoritmo evolutivo multiobjetivo (MOEA/D-DE)

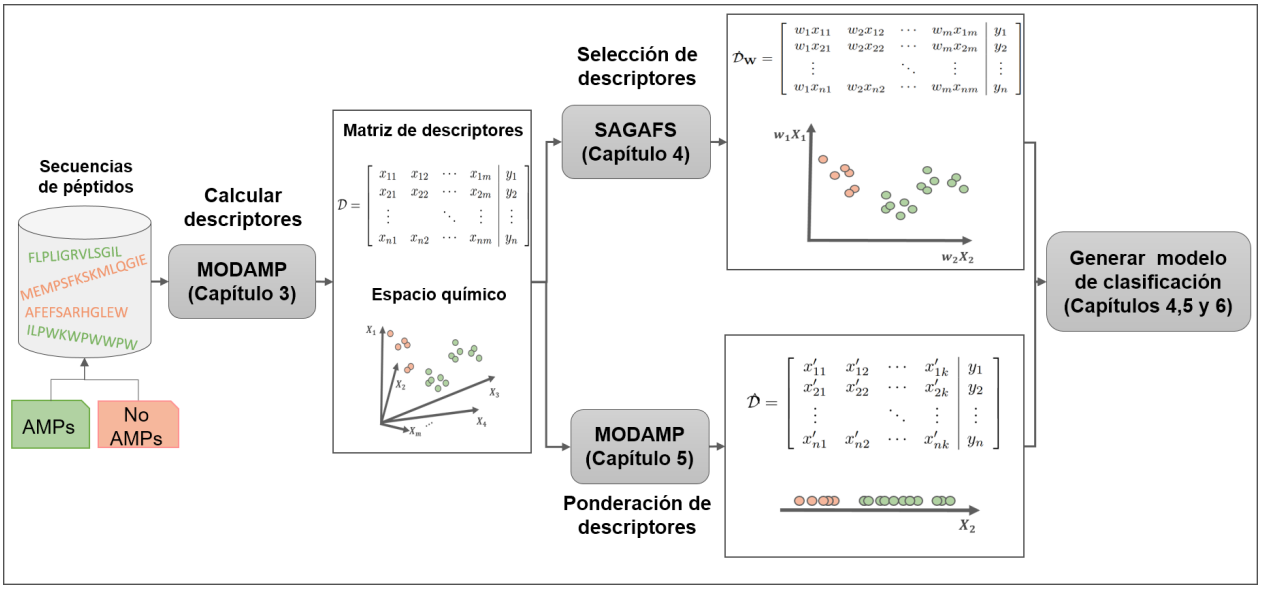


Figura 38. Resumen gráfico del problema de selección de descriptores moleculares para la clasificación de la actividad antimicrobiana.

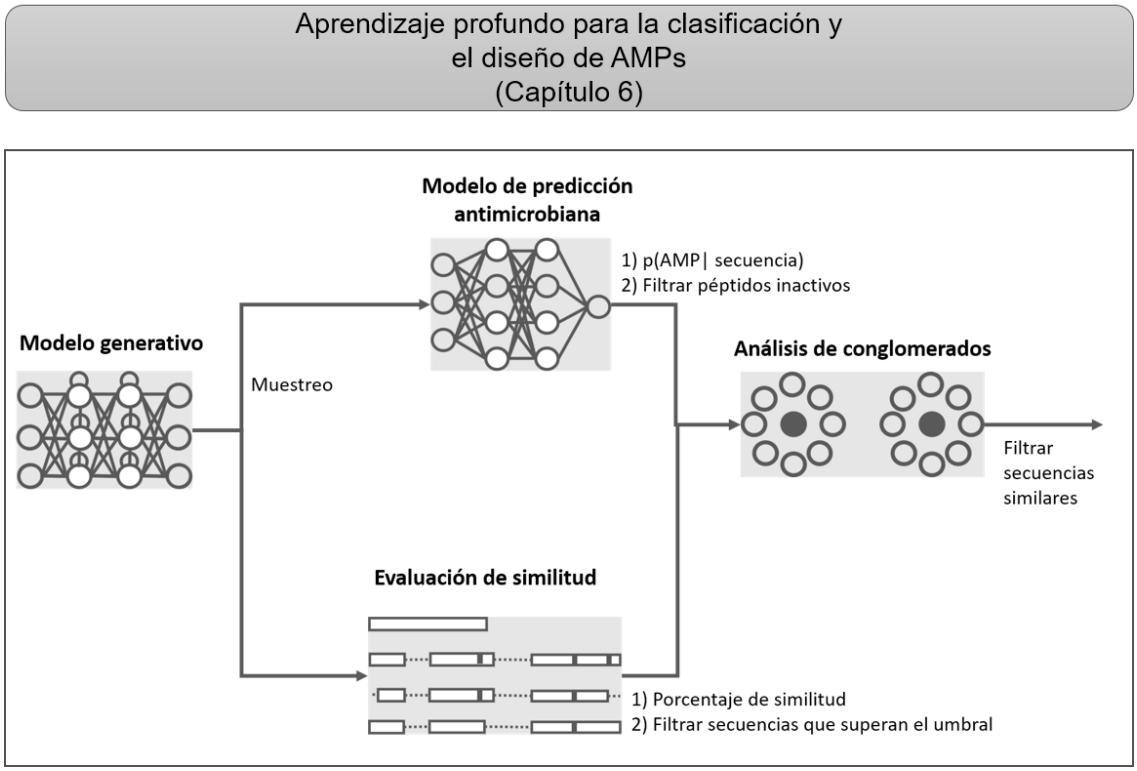


Figura 39. Resumen gráfico del problema de diseño de AMPs.

(Zhang y Li, 2007; Li y Zhang, 2009) (Ver Capítulo 5). Dado que la solución al problema no es única sino dada por un conjunto de soluciones en conflicto entre sí, introducimos un enfoque de toma de decisiones multicriterio para seleccionar las soluciones con diferentes grados de satisfacción entre las distancias intra-clase y entre-clases para la clase objetivo. Luego, en el espacio transformado se construyó un modelo de clasificación binaria. Los resultados más importantes con este enfoque fueron los siguientes:

- La solución con la que se obtiene mejor desempeño de clasificación es aquella en la que la distancia intra-clase para la clase de interés y la distancia entre-clases tienen el mismo peso.
- El análisis de los resultados experimentales, en seis conjuntos de datos no balanceados, indica que la metodología propuesta es eficaz en el desarrollo de modelos para predecir los péptidos antimicrobianos. En particular, en la generación de modelos de discriminación contra una actividad antimicrobiana específica, como la antibacteriana.
- En lo que respecta a la clasificación de péptidos cuyo origen y destino son bacterias (bacteriocinas) no se logra tener un buen desempeño. Conjeturamos que una posible explicación de este resultado es el menor número de casos para este tipo de actividad, dado que entre las bases de datos DAMPD y APD3, se obtiene un mejor resultado para la de APD3 que es la que tiene un mayor número de bacteriocinas.
- Como aplicaciones prácticas, este enfoque se puede aplicar a un conjunto de datos biológicos con características similares al problema de los péptidos antimicrobianos, donde los casos positivos tienen actividades similares, mientras que los casos negativos no necesariamente.
- Nuestro enfoque se puede combinar con otros que utilizan un sesgo hacia un algoritmo de aprendizaje de máquina en particular para elegir la representación.

La tercera contribución de este trabajo de tesis es con respecto a la selección de descriptores moleculares usando un enfoque de envoltura, en donde proponemos un algoritmo evolutivo novedoso y eficaz para el problema de selección de características en la identificación de péptidos antimicrobianos (ver Capítulo 4). El enfoque combina

dos algoritmos, un algoritmo genético basado en especies y un algoritmo evolutivo de longitud variable. Además, como función objetivo se usa la medida de MCC de un clasificador inducido y un término que depende del número de descriptores moleculares seleccionados. La solución encontrada por nuestro enfoque es la de tamaño mínimo entre las soluciones exploradas, esto gracias a la restricción de tamaño incluida en la función objetivo. Los resultados más importantes de este enfoque fueron los siguientes:

- Los experimentos computacionales muestran que el método propuesto es capaz de encontrar una representación para los péptidos capaces de generar modelos que superan a los métodos de última generación que están disponibles públicamente para la predicción del AMP.
- Nuestro enfoque permite reducir en promedio en un 90 % el número de descriptores moleculares cuando se utiliza RF.
- Este enfoque es muy costoso computacionalmente debido a que se tiene que construir un clasificador por cada solución, sin embargo, una vez que se encuentra la solución óptima, esta queda fija para entrenar el modelo y predecir nuevos péptidos antimicrobianos.

La última contribución es con respecto al diseño de nuevas secuencias con actividad antimicrobiana potencial, en este trabajo se propone una metodología con el objetivo de generar secuencias activas y diferentes a un conjunto de secuencias de AMPs (ver Capítulo 6). Para este propósito, se propone un enfoque compuesto por cuatro componentes principales: el primero un modelo generativo para el diseño de péptidos a nivel de aminoácidos; el segundo, un modelo generado por una red neuronal óptima profunda en el espacio químico, esta tiene como objetivo determinar si los péptidos generados son antimicrobianos; el tercer componente es una función de evaluación de similitud, para eliminar los péptidos diseñados que son similares a nivel de secuencias de los AMPs existentes; por último, un componente de análisis de conglomerados para agrupar las secuencias diseñadas con un porcentaje de similitud y seleccionar un representante por conglomerado. Los resultados preliminares de este enfoque son los siguientes:

- La red neuronal profunda óptima genera un modelo con una excelente capacidad para aprender qué es un péptido antimicrobiano en el espacio químico. Este resultado es importante de resaltar debido al gran número de secuencias con las que se entrenó y probó el modelo, *i.e.*, 43403 secuencias para entrenamiento y 8681 para prueba, respectivamente. Comparando los conjuntos que se han utilizado para entrenar y probar, el que utilizamos es el más significativo que hasta donde tenemos conocimiento se ha utilizado para la clasificación de AMPs. Por ejemplo, el modelo de CAMP en (Thomas *et al.*, 2009) utiliza 4915 secuencias para entrenar y 2106 para prueba. El modelo de clasificación de la actividad antimicrobiana tiene un desempeño de 0.97 en la medida de área bajo la curva ROC (AUC) para la validación cruzada de 5 pliegues y para la prueba, respectivamente.
- En lo que respecta al modelo generativo, este es capaz de producir secuencias que el modelo de clasificación identifica con alta probabilidad como AMPs, sin embargo, todavía los resultados de este modelo son preliminares y se necesitan de más pruebas para descartar *overfitting* o algún otro comportamiento extraño. Además, para tener certeza de que los péptidos tengan la actividad antimicrobiana se requiere realizar los experimentos *in vitro* correspondientes.

En resumen, nuestras contribuciones son direccionadas a seleccionar una representación óptima del espacio químico en donde se lleva a cabo la clasificación binaria de la actividad antimicrobiana. Además, generamos un modelo de clasificación de la actividad antimicrobiana que supera a los métodos accesibles desde la web de CAMPR3 (Waghu *et al.*, 2014) bajo el mismo conjunto de prueba. En lo que respecta al conjunto de datos, intentamos crear un conjunto grande no redundante, en donde analizamos y combinamos los conjuntos de datos negativos del estado del arte.

7.2. Trabajo futuro

En las siguientes subsecciones se presentan algunas ideas de proyectos de investigación como trabajo futuro.

7.2.1. Ponderación de descriptores moleculares

Dado que, para el caso de actividad antimicrobiana específica, el modelo que se construyó obtuvo el mejor desempeño, entonces la futura dirección de esta investigación tiene como objetivo la construcción de clasificadores especializados en actividades antimicrobianas específicas como antivirales, antifúngicos, antitumorales, entre otros. Lo anterior es con el objetivo de mostrar que nuestro enfoque funciona para actividades antimicrobianas específicas.

Por otra parte, nuestro enfoque se encuentra limitado al número de soluciones que elegimos del conjunto aproximado de Pareto para construir los modelos de clasificación binaria. Por este motivo, puede ocurrir que en las soluciones no seleccionadas se encuentre una solución mejor que las seleccionadas por nuestro enfoque de toma de decisiones multicriterio. En esta dirección, como trabajo futuro queda mejorar el enfoque de toma de decisiones para asegurar siempre elegir la solución que genere el mejor modelo de clasificación dada una medida de desempeño.

7.2.2. Selección de los descriptores moleculares

En este enfoque, la representación óptima de los péptidos en el espacio químico se encuentra sesgada al algoritmo de aprendizaje de máquina que se seleccione. Es importante señalar que nuestros hallazgos están limitados por el tamaño y la base de datos donde se probó. Esto es, el mejor modelo encontrado para un conjunto de datos no es comparable para otro modelo con otra base de datos. Como trabajo futuro para este enfoque considero importante trabajar en los siguientes puntos: primero, desde el punto de vista de los enfoques evolutivos queda trabajar la comparación de este enfoque con otros enfoques similares (*e.g.,* algoritmo genético basado en islas) en donde la búsqueda de la solución óptima es heterogénea y se divide el espacio en subespacios de acuerdo con el tamaño de los subconjuntos. Segundo, desde el enfoque de problema de clasificación queda explorar si la búsqueda de la solución eficiente para el problema de clasificación binaria de la actividad antimicrobiana se mantiene para el problema de clasificación multiclase y multietiqueta. Tercero, comparar el tipo de representación que se obtiene al entrenar con secuencias que sólo contienen al péptido maduro con aquellas que incluyen la región pro y prepéptido, respectivamente. Lo

anterior, con el objetivo de analizar qué tanta variabilidad hay en las representaciones que se seleccionan en el espacio químico, así como en el desempeño de los modelos que se generan con estas representaciones.

7.2.3. Diseño de nuevos AMPs

Algunas de las limitaciones del enfoque que proponemos es que el modelo generativo no garantiza generar solo secuencias que sean antimicrobianas y el modelo de clasificación actúa de forma separada al modelo generativo. Por lo anterior, no existe retroalimentación entre el modelo de clasificación hacia el modelo generativo. En esta dirección, como trabajo futuro se propone unir ambos modelos a través de las redes generativas adversarias con aprendizaje por refuerzo. Esta modificación a nuestro enfoque permitirá sesgar el espacio de búsqueda hacia secuencias que pertenecen a regiones inexploradas donde nuevos péptidos antimicrobianos con propiedades deseables se encuentren.

Literatura citada

Abido, M. (2003). A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. *Electric power systems research*, **65**(1): 71–81.

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: towards an organized collection enabling visual network analysis. *Bioinformatics*.

Aha, D. y Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*, **6**: 37–66.

Ali, S. I. y Shahzad, W. (2012). A feature subset selection method based on conditional mutual information and ant colony optimization. *methods*, **1**(2): 3–4.

Amaldi, E. y Kann, V. (1998a). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, **209**(1-2): 237–260.

Amaldi, E. y Kann, V. (1998b). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, **209**(1): 237–260.

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

Baldi, P. y Brunak, S. (2000). Bioinformatics - the machine learning approach. En: *SIGA*.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., y Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**(5): 412–424.

Beltran, J. A. y Brizuela, C. A. (2016). Design of selective cationic antibacterial peptides: A multiobjective genetic algorithm approach. En: *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 484–491.

Beltrán, J. A., Aguilera-Mendoza, L., y Brizuela, C. A. (2017). Feature weighting for antimicrobial peptides classification: A multi-objective evolutionary approach. En: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 276–283.

Beltran, J. A., Aguilera-Mendoza, L., y Brizuela, C. A. (2018). Optimal selection of molecular descriptors for antimicrobial peptides classification: an evolutionary feature weighting approach. *BMC genomics*, **19**(7): 672.

Benhenda, M. (2017). Chemgan challenge for drug discovery: can ai reproduce natural chemical diversity? *arXiv preprint arXiv:1708.08227*.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., y Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, **28**(1): 235–242.

Boman, H. (2003). Antibacterial peptides: basic facts and emerging concepts. *Journal of internal medicine*, **254**(3): 197–215.

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

Breunig, M. M., Kriegel, H.-P., Ng, R. T., y Sander, J. (2000). Lof: identifying density-based local outliers. En: *ACM sigmod record*. ACM, Vol. 29, pp. 93–104.

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

Brown, E. D. y Wright, G. D. (2016). Antibacterial drug discovery in the resistance era. *Nature*, **529**(7586): 336–343.

Cai, C., Gong, J., Liu, X., Gao, D., y Li, H. (2013). Molecular similarity: methods and performance. *Chinese Journal of Chemistry*, **31**(9): 1123–1132.

Chen, Y., Mant, C. T., Farmer, S. W., Hancock, R. E., Vasil, M. L., y Hodges, R. S. (2005). Rational design of α-helical antimicrobial peptides with enhanced activities and specificity/therapeutic index. *Journal of biological chemistry*, **280**(13): 12316–12329.

Cherkasov, A., Hilpert, K., Jenssen, H., Fjell, C. D., Waldbrook, M., Mullaly, S. C., Volkmer, R., y Hancock, R. E. (2008a). Use of artificial intelligence in the design of small peptide antibiotics effective against a broad spectrum of highly antibiotic-resistant superbugs. *ACS chemical biology*, **4**(1): 65–74.

Cherkasov, A., Hilpert, K., Jenssen, H., Fjell, C. D., Waldbrook, M., Mullaly, S. C., Volkmer, R., y Hancock, R. E. (2008b). Use of artificial intelligence in the design of small peptide antibiotics effective against a broad spectrum of highly antibiotic-resistant superbugs. *ACS chemical biology*, **4**(1): 65–74.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., *et al.* (2007). *Evolutionary algorithms for solving multi-objective problems*, Vol. 5. Springer. New York.

Consortium, U. *et al.* (2018). Uniprot: the universal protein knowledgebase. *Nucleic acids research*, **46**(5): 2699.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. En: *ICML*. Citeseer, Vol. 1, pp. 74–81.

Dash, M. y Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, **1**(3): 131–156.

Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, **6**(2): 182–197.

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

Dhillon, J., Parti, S., y Kothari, D. (1993). Stochastic economic emission load dispatch. *Electric Power Systems Research*, **26**(3): 179–186.

Dubchak, I., Muchnik, I., Holbrook, S. R., y Kim, S.-H. (1995). Prediction of protein folding class using global description of amino acid sequence. *Proceedings of the National Academy of Sciences*, **92**(19): 8700–8704.

Eiben, A. E., Smith, J. E., *et al.* (2003). *Introduction to evolutionary computing*, Vol. 53. Springer.

Eisenberg, D., Weiss, R. M., Terwilliger, T. C., y Wilcox, W. (1982). Hydrophobic moments and protein structure. En: *Faraday Symposia of the Chemical Society*. Royal Society of Chemistry, Vol. 17, pp. 109–120.

Eisenberg, D., Schwarz, E., Komaromy, M., y Wall, R. (1984). Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *Journal of molecular biology*, **179**(1): 125–142.

Eliseev, I. E., Terterov, I. N., Yudenko, A. N., y Shamova, O. V. (2018). Linking sequence patterns and functionality of alphahelical antimicrobial peptides. *Bioinformatics*.

Emmanouilidis, C., Hunter, A., y MacIntyre, J. (2000). A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. En: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. IEEE, Vol. 1, pp. 309–316.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., y Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, **9**(Aug): 1871–1874.

Fernandes, F. C., Rigden, D. J., y Franco, O. L. (2012). Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application. *Peptide Science*, **98**(4): 280–287.

Fernandez-Escamilla, A.-M., Rousseau, F., Schymkowitz, J., y Serrano, L. (2004). Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins. *Nature biotechnology*, **22**(10): 1302–1306.

Fields, F. R., Carothers, K. E., Balsara, R. D., Ploplis, V. A., Castellino, F. J., y Lee, S. W. (2018). Rational design of syn-safencin, a novel linear antimicrobial peptide derived from the circular bacteriocin safencin as-48. *The Journal of antibiotics*, p. 1.

Fischbach, M. A. y Walsh, C. T. (2009). Antibiotics for emerging pathogens. *Science*, **325**(5944): 1089–1093.

Fjell, C. D., Jenssen, H., Hilpert, K., Cheung, W. A., Pante, N., Hancock, R. E., y Cherkasov, A. (2009). Identification of novel antibacterial peptides by chemoinformatics and machine learning. *Journal of medicinal chemistry*, **52**(7): 2006–2015.

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.

Fox, J. L. (2013). Antimicrobial peptides stage a comeback. *Nature Biotechnology*, **31**: 379.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, **11**(1): 86–92.

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

Gasteiger, E., Hoogland, C., Gattiker, A., Duvaud, S., Wilkins, M. R., Appel, R. D., y Bairoch, A. (2005). *Protein identification and analysis tools on the ExPASy server*. Springer.

Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press.

Guruprasad, K., Reddy, B. B., y Pandit, M. W. (1990). Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Engineering, Design and Selection*, **4**(2): 155–161.

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

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., y Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, **11**(1): 10–18.

Hamid, M. N. y Friedberg, I. (2018). Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *BioRxiv*, p. 255505.

Hammami, R., Zouhir, A., Hamida, J. B., y Fliss, I. (2007). Bactibase: a new web-accessible database for bacteriocin characterization. *Bmc Microbiology*, **7**(1): 89.

Hampton, T. (2013). Report reveals scope of us antibiotic resistance threat. *Jama*, **310**(16): 1661–1663.

Hancock, R. E. y Diamond, G. (2000). The role of cationic antimicrobial peptides in innate host defences. *Trends in microbiology*, **8**(9): 402–410.

Helguera, A. M., Combes, R. D., González, M. P., y Cordeiro, M. (2008). Applications of 2d descriptors in drug design: a dragon tale. *Current topics in medicinal chemistry*, **8**(18): 1628–1655.

Hertel, L., Collado, J., Sadowski, P., y Baldi, P. (2018). Sherpa: Hyperparameter optimization for machine learning models.

Hilpert, K., Winkler, D. F., y Hancock, R. E. (2007). Peptide arrays on cellulose support: Spot synthesis, a time and cost efficient method for synthesis of large numbers of peptides in a parallel and addressable fashion. *Nature protocols*, **2**(6): 1333.

Hilpert, K., Fjell, C. D., y Cherkasov, A. (2008). Short linear cationic antimicrobial peptides: screening, optimizing, and prediction. En: *Peptide-Based Drug Design*. Springer, pp. 127–159.

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

Hocke, J. y Martinetz, T. (2015). Maximum distance minimization for feature weighting. *Pattern Recognition Letters*, **52**: 48–52.

Horn, R. A., Horn, R. A., y Johnson, C. R. (1990). *Matrix analysis*. Cambridge university press.

Hsu, H.-H., Hsieh, C.-W., y Lu, M.-D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, **38**(7): 8144–8150.

Huang, J., Cai, Y., y Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, **28**(13): 1825–1844.

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.

Hughes, A. B. (2013). *Amino acids, peptides and proteins in organic chemistry, analysis and function of amino acids and peptides*, Vol. 5. John Wiley & Sons.

Hunninghake, G. W. y Gadek, J. E. (1995). The alveloar macrophage. En: T. J. R. Harris (ed.), *Cultured Human Cells and Tissues*. Academic Press, New York, pp. 54–56. Stoner G (Series Editor): Methods and Perspectives in Cell Biology, vol 1.

Ikai, A. (1980). Thermostability and aliphatic index of globular proteins. *The Journal of Biochemistry*, **88**(6): 1895–1898.

James, G., Witten, D., Hastie, T., y Tibshirani, R. (2013a). *An introduction to statistical learning*, Vol. 112. Springer.

James, G., Witten, D., Hastie, T., y Tibshirani, R. (2013b). Linear model selection and regularization. En: *An Introduction to Statistical Learning*. Springer, pp. 203–264.

Jenssen, H. (2011). Descriptors for antimicrobial peptides. *Expert opinion on drug discovery*, **6**(2): 171–184.

John, G. H., Kohavi, R., Pfleger, K., *et al.* (1994). Irrelevant features and the subset selection problem. En: *Machine learning: proceedings of the eleventh international conference*. pp. 121–129.

Kabir, M. M., Shahjahan, M., y Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing*, **74**(17): 2914–2928.

Käll, L., Krogh, A., y Sonnhammer, E. L. (2007). Advantages of combined transmembrane topology and signal peptide prediction¯the phobius web server. *Nucleic acids research*, **35**(suppl 2): W429–W432.

Kang, H.-K., Kim, C., Seo, C. H., y Park, Y. (2017). The therapeutic applications of antimicrobial peptides (amps): a patent review. *Journal of microbiology*, **55**(1): 1–12.

Karp, G. (2009). *Cell and molecular biology: concepts and experiments*. John Wiley & Sons.

Kawashima, S. y Kanehisa, M. (2000). Aaindex: amino acid index database. *Nucleic acids research*, **28**(1): 374–374.

Kim, I.-W., Lee, J. H., Subramaniyam, S., Yun, E.-Y., Kim, I., Park, J., y Hwang, J. S. (2016). De novo transcriptome analysis and detection of antimicrobial peptides of the american cockroach periplaneta americana (linnaeus). *PloS one*, **11**(5): e0155304.

Kingma, D. P. y Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kleandrova, V. V., Ruso, J. M., Speck-Planche, A., y Dias Soeiro Cordeiro, M. N. (2016). Enabling the discovery and virtual screening of potent and safe antimicrobial peptides. simultaneous prediction of antibacterial activity and cytotoxicity. *ACS combinatorial science*, **18**(8): 490–498.

Klein, P., Kanehisa, M., y DeLisi, C. (1984). Prediction of protein function from sequence properties: Discriminant analysis of a data base. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, **787**(3): 221–226.

Kohavi, R. (1995). *Wrappers for performance enhancement and obvious decision graphs*. Tesis de doctorado, Stanford University, Computer Science Department.

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

Kozlowski, L. P. (2016). Ipc–isoelectric point calculator. *Biology direct*, **11**(1): 55.

Lata, S., Sharma, B., y Raghava, G. (2007). Analysis and prediction of antibacterial peptides. *BMC bioinformatics*, **8**(1): 263.

Lata, S., Mishra, N. K., y Raghava, G. P. (2010). Antibp2: improved version of antibacterial peptide prediction. *BMC bioinformatics*, **11**(1): S19.

Li, H. y Zhang, Q. (2009). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, **13**(2): 284–302.

Li, Z.-R., Lin, H. H., Han, L., Jiang, L., Chen, X., y Chen, Y. Z. (2006). Profeat: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Research*, **34**(suppl_2): W32–W37.

Lichman, M. (2013). UCI machine learning repository.

Lin, E. y Lane, H. R. (2017). Machine learning and systems genomics approaches for multi-omics data. En: *Biomarker Research*.

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

Linding, R., Schymkowitz, J., Rousseau, F., Diella, F., y Serrano, L. (2004a). A comparative study of the relationship between protein structure and β-aggregation in globular and intrinsically disordered proteins. *Journal of molecular biology*, **342**(1): 345–353.

Linding, R., Schymkowitz, J., Rousseau, F., Diella, F., y Serrano, L. (2004b). A comparative study of the relationship between protein structure and β-aggregation in globular and intrinsically disordered proteins. *Journal of molecular biology*, **342**(1): 345–353.

Lira, F., Perez, P. S., Baranauskas, J. A., y Nozawa, S. R. (2013). Prediction of antimicrobial activity of synthetic peptides by a decision tree model. *Appl. Environ. Microbiol.*, **79**(10): 3156–3159.

Litjens, G. J. S., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J., van Ginneken, B., y Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, **42**: 60–88.

Liu, H. y Motoda, H. (1998). *Feature extraction, construction and selection: A data mining perspective*, Vol. 453. Springer Science & Business Media.

Liu, Z., Bensmail, H., y Tan, M. (2012). Efficient feature selection and multiclass classification with integrated instance and model based learning. *Evolutionary Bioinformatics*, **8**: 197.

Loose, C., Jensen, K., Rigoutsos, I., y Stephanopoulos, G. (2006). A linguistic model for the rational design of antimicrobial peptides. *Nature*, **443**(7113): 867.

Maaten, L. v. d. y Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, **9**(Nov): 2579–2605.

Mahlapuu, M., Håkansson, J., Ringstad, L., y Björn, C. (2016). Antimicrobial peptides: an emerging category of therapeutic agents. *Frontiers in cellular and infection microbiology*, **6**.

Mannhold, R., Kubinyi, H., y Folkers, G. (2011). *Virtual screening: principles, challenges, and practical guidelines*, Vol. 48. John Wiley & Sons.

Matsuzaki, K. (2019). *Antimicrobial Peptides, Advances in Experimental Medicine and Biology*. 1117. Springer.

Miettinen, K. (2012). *Nonlinear multiobjective optimization*, Vol. 12. Springer Science & Business Media.

Müller, A. T., Hiss, J. A., y Schneider, G. (2018). Recurrent neural network model for constructive peptide design. *Journal of chemical information and modeling*, **58**(2): 472–479.

Munita, J. M. y Arias, C. A. (2016). Mechanisms of antibiotic resistance. *Microbiology spectrum*, **4**(2).

Murphy, L. R., Wallqvist, A., y Levy, R. M. (2000). Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein Engineering*, **13**(3): 149–152.

Nagarajan, D., Nagarajan, T., Roy, N., Kulkarni, O., Ravichandran, S., Mishra, M., Chakravortty, D., y Chandra, N. (2018). Computational antimicrobial peptide design and evaluation against multidrug-resistant clinical isolates of bacteria. *Journal of Biological Chemistry*, **293**(10): 3492–3509.

Neill, J. (2016). Tackling drug-resistant infections globally: Final report and recommendations-the review on antimicrobial resistance. *Date of access*, **16**(09).

Neme, R., Amador, C., Yildirim, B., McConnell, E., y Tautz, D. (2017). Random sequences are an abundant source of bioactive rnas or peptides. *Nature ecology & evolution*, **1**(6): 0127.

Olivecrona, M., Blaschke, T., Engkvist, O., y Chen, H. (2017). Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, **9**(1): 48.

Pal, D. y Chakrabarti, P. (2000). Conformational similarity indices between different residues in proteins and α-helix propensities. *Journal of Biomolecular Structure and Dynamics*, **18**(2): 273–280.

Paul, S. y Das, S. (2015). Simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach. *Pattern Recognition Letters*, **65**: 51–59.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.* (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, **12**(Oct): 2825–2830.

Petasis, G. (2012). Machine learning in natural language processing.

Piotto, S. P., Sessa, L., Concilio, S., y Iannelli, P. (2012). Yadamp: yet another database of antimicrobial peptides. *International journal of antimicrobial agents*, **39**(4): 346–351.

Porto, W. F., Fernandes, F. C., y Franco, O. L. (2010). An svm model based on physicochemical properties to predict antimicrobial activity from protein sequences with cysteine knot motifs. En: *Brazilian Symposium on Bioinformatics*. Springer, pp. 59–62.

Porto, W. F., Fensterseifer, I. C., Ribeiro, S. M., y Franco, O. L. (2018a). Joker: An algorithm to insert patterns into sequences for designing antimicrobial peptides. *Biochimica et Biophysica Acta (BBA)-General Subjects*, **1862**(9): 2043–2052.

Porto, W. F., Irazazabal, L., Alves, E. S., Ribeiro, S. M., Matos, C. O., Pires, Á. S., Fensterseifer, I. C., Miranda, V. J., Haney, E. F., Humblot, V., *et al.* (2018b). In silico optimization of a guava antimicrobial peptide enables combinatorial exploration for peptide design. *Nature communications*, **9**(1): 1490.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

R. Hansen, P. (2017). *Antimicrobial Peptides: Methods and Protocols*, Vol. 1548.

Randou, E. G., Veltri, D., y Shehu, A. (2013). Binary response models for recognition of antimicrobial peptides. En: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*. ACM, p. 76.

Raventos, D., Taboureau, O., Mygind, P., Nielsen, J., Sonksen, C., y Kristensen, H.-H. (2005a). Improving on nature's defenses: optimization & high throughput screening of antimicrobial peptides. *Combinatorial chemistry & high throughput screening*, **8**(3): 219–233.

Raventos, D., Taboureau, O., Mygind, P., Nielsen, J., Sonksen, C., y Kristensen, H.-H. (2005b). Improving on nature's defenses: optimization & high throughput screening of antimicrobial peptides. *Combinatorial chemistry & high throughput screening*, **8**(3): 219–233.

Robinson, J. A. (2011). Protein epitope mimetics as anti-infectives. *Current opinion in chemical biology*, **15**(3): 379–386.

Rondón-Villarreal, P., Sierra, D. A., y Torres, R. (2014). Classification of antimicrobial peptides by using the p-spectrum kernel and support vector machines. En: *Advances in Computational Biology*. Springer, pp. 155–160.

Rose, G. D., Geselowitz, A. R., Lesser, G. J., Lee, R. H., y Zehfus, M. H. (1985). Hydrophobicity of amino acid residues in globular proteins. *Science*, **229**(4716): 834–838.

Rousseau, F., Schymkowitz, J., y Serrano, L. (2006). Protein aggregation and amyloidosis: confusion of the kinds? *Current opinion in structural biology*, **16**(1): 118–126.

Roy, K., Kar, S., y Das, R. N. (2015). *QSAR/QSPR Modeling: Introduction*, pp. 1–36. Springer International Publishing, Cham.

Rozek, T., Wegener, K. L., Bowie, J. H., Olver, I. N., Carver, J. A., Wallace, J. C., y Tyler, M. J. (2000). The antibiotic and anticancer active aurein peptides from the australian bell frogs litoria aurea and litoria raniformis: the solution structure of aurein 1.2. *European Journal of Biochemistry*, **267**(17): 5330–5341.

Ruiz-Blanco, Y. B., Paz, W., Green, J., y Marrero-Ponce, Y. (2015). Protdcal: A program to compute general-purpose-numerical descriptors for sequences and 3d-structures of proteins. *BMC bioinformatics*, **16**(1): 1.

Sánchez-Gómez, S., Japelj, B., Jerala, R., Moriyón, I., Alonso, M. F., Leiva, J., Blondelle, S. E., Andrä, J., Brandenburg, K., Lohner, K., *et al.* (2011). Structural features governing the activity of lactoferricin-derived peptides that act in synergy with antibiotics against pseudomonas aeruginosa in vitro and in vivo. *Antimicrobial agents and chemotherapy*, **55**(1): 218–228.

Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L., y Aspuru-Guzik, A. (2017). Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic). *ChemRxiv*.

Scannell, J. W., Blanckley, A., Boldon, H., y Warrington, B. (2012). Diagnosing the decline in pharmaceutical r&d efficiency. *Nature reviews Drug discovery*, **11**(3): 191.

Schneider, P., Müller, A. T., Gabernet, G., Button, A. L., Posselt, G., Wessler, S., Hiss, J. A., y Schneider, G. (2017). Hybrid network model for “deep learning” of chemical data: application to antimicrobial peptides. *Molecular informatics*, **36**(1-2).

Sebe, N., Cohen, I., Garg, A., y Huang, T. S. (2005). Machine learning in computer vision. En: *Computational Imaging and Vision*.

Segler, M. H., Kogej, T., Tyrchan, C., y Waller, M. P. (2017). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*.

Seshadri Sundararajan, V., Gabere, M. N., Pretorius, A., Adam, S., Christoffels, A., Lehväslaiho, M., Archer, J. A., y Bajic, V. B. (2011). Dampd: a manually curated antimicrobial peptide database. *Nucleic acids research*, **40**(D1): D1108–D1112.

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. En: *Proceedings of the eleventh international conference on machine learning*. pp. 293–301.

Smullen, D., Gillett, J., Heron, J., y Rahnamayan, S. (2014). Genetic algorithm with self-adaptive mutation controlled by chromosome similarity. En: *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, pp. 504–511.

Somol, P., Pudil, P., y Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on pattern analysis and machine intelligence*, **26**(7): 900–912.

Stahura, F. L. y Bajorath, J. (2003). Partitioning methods for the identification of active molecules. *Current medicinal chemistry*, **10**(8): 707–715.

Stracuzzi, D. J. (2007). Randomized feature selection. *Computational Methods of Feature Selection*, pp. 41–62.

Taber, H. W. (2001). Introduction to the peptide antibiotics. En: *Peptide Antibiotics*. CRC Press, pp. 7–16.

Tacconelli, E., Carrara, E., Savoldi, A., Harbarth, S., Mendelson, M., Monnet, D. L., Pulcini, C., Kahlmeter, G., Kluytmans, J., Carmeli, Y., *et al.* (2018). Discovery, research, and development of new antibiotics: the who priority list of antibiotic-resistant bacteria and tuberculosis. *The Lancet Infectious Diseases*, **18**(3): 318–327.

Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K., y Idicula-Thomas, S. (2009). Camp: a useful resource for research on antimicrobial peptides. *Nucleic acids research*, p. gkp1021.

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

Todeschini, R. y Consonni, V. (2009). *Molecular Descriptors for Chemoinformatics, Volume 41 (2 Volume Set)*, Vol. 41. John Wiley & Sons.

Tomii, K. y Kanehisa, M. (1996). Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Engineering, Design and Selection*, **9**(1): 27–36.

Torrent, M., Andreu, D., Nogués, V. M., y Boix, E. (2011). Connecting peptide physicochemical and antimicrobial properties by a rational prediction model. *PloS one*, **6**(2): e16968.

Tossi, A., Tarantino, C., y Romeo, D. (1997). Design of synthetic antimicrobial peptides based on sequence analogy and amphipathicity. *European journal of biochemistry*, **250**(2): 549–558.

Tossi, A., Sandri, L., y Giangaspero, A. (2003). New consensus hydrophobicity scale extended to non-proteinogenic amino acids. En: *27th European Peptide Symposium*. Edizioni Ziino, pp. 416–417.

Tucker, A. T., Leonard, S. P., DuBois, C. D., Knauf, G. A., Cunningham, A. L., Wilke, C. O., Trent, M. S., y Davies, B. W. (2018). Discovery of next-generation antimicrobials through bacterial self-screening of surface-displayed peptide libraries. *Cell*.

van Heel, A. J., de Jong, A., Montalban-Lopez, M., Kok, J., y Kuipers, O. P. (2013). Bagel3: automated identification of genes encoding bacteriocins and (non-) bactericidal posttranslationally modified peptides. *Nucleic acids research*, **41**(W1): W448–W453.

Veltri, D., Kamath, U., y Shehu, A. (2017). Improving recognition of antimicrobial peptides and target selectivity through machine learning and genetic programming. *IEEE/ACM transactions on computational biology and bioinformatics*, **14**(2): 300–313.

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

Waghu, F. H., Gopi, L., Barai, R. S., Ramteke, P., Nizami, B., y Idicula-Thomas, S. (2014). Camp: Collection of sequences and structures of antimicrobial peptides. *Nucleic acids research*, **42**(D1): D1154–D1158.

Wang, G., Li, X., y Wang, Z. (2009). Apd2: the updated antimicrobial peptide database and its application in peptide design. *Nucleic acids research*, **37**(suppl 1): D933–D937.

Wang, G., Li, X., Zasloff, M., *et al.* (2010). A database view of naturally occurring antimicrobial peptides: nomenclature, classification and amino acid sequence analysis. *Antimicrobial peptides: discovery, design and novel therapeutic strategies*, pp. 1–21.

Wang, G., Li, X., y Wang, Z. (2016). Apd3: the antimicrobial peptide database as a tool for research and education. *Nucleic Acids Research*, **44**(D1): D1087–D1093.

Wang, J., Dou, X., Song, J., Lyu, Y., Zhu, X., Xu, L., Li, W., y Shan, A. (2019). Antimicrobial peptides: Promising alternatives in the post feeding antibiotic era. *Medicinal Research Reviews*, **39**(3): 831–859.

Webb, A. R. (2003). *Statistical pattern recognition*, capítulo 9, pp. 305–360. John Wiley & Sons, segunda edición.

Witten, I. H., Frank, E., Hall, M. A., y Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

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.

Yap, C. W. (2011). Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, **32**(7): 1466–1474.

Yeaman, M. R. y Yount, N. Y. (2003). Mechanisms of antimicrobial peptide action and resistance. *Pharmacological reviews*, **55**(1): 27–55.

Zasloff, M. (1987). Magainins, a class of antimicrobial peptides from xenopus skin: isolation, characterization of two active forms, and partial cdna sequence of a precursor. *Proceedings of the National Academy of Sciences*, **84**(15): 5449–5453.

Zavascki, A. P., Goldani, L. Z., Li, J., y Nation, R. L. (2007). Polymyxin b for the treatment of multidrug-resistant pathogens: a critical review. *Journal of antimicrobial chemotherapy*, **60**(6): 1206–1215.

Zebulum, R. S., Vellasco, M., y Pacheco, M. A. (2000). Variable length representation in evolutionary electronics. *Evolutionary Computation*, **8**(1): 93–120.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, L.-j. y Gallo, R. L. (2016). Antimicrobial peptides. *Current Biology*, **26**(1): R14–R19.

Zhang, Q. y Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, **11**(6): 712–731.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., y Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, **7**(2): 117–132.

Apéndice A

A.1. MODAMP: cálculo de los descriptores moleculares de dos péptidos

En esta sección se ilustra un ejemplo del cómputo de los descriptores moleculares para las secuencias de péptidos con identificador DRAMP16957 y DBAASP5756. MODAMP recibe como entrada un archivo fasta (ver Tabla 21) y entrega como salida un archivo CSV con el cómputo de los descriptores moleculares (ver Tabla 22).

Tabla 21. Ejemplo de archivo de entrada para MODAMP. MODAMP recibe como entrada un archivo fasta con secuencias de aminoácidos validas a las cuales se les calculará los descriptores moleculares.

```
>DRAMP16957
IKKEIEAIKKEQEAIKKLLQLTVWGIKQLQARIL
>DBAASP5756
MAQKIISTIGKLPKWIIKTVNKFTKK
```

Tabla 22. Ejemplo de archivo de salida para MODAMP. MODAMP entrega como salida un archivo CSV con las secuencias transformadas en valores de descriptores moleculares.

ID	DRAMP16957	DBAASP5756
C_all(A)	8.82353	3.84615
C_all(C)	0	0
C_all(D)	0	0
C_all(E)	11.76471	0
C_all(F)	0	3.84615
C_all(G)	2.94118	3.84615
C_all(H)	0	0
C_all(I)	17.64706	19.23077
C_all(K)	20.58824	26.92308
C_all(L)	14.70588	3.84615
C_all(M)	0	3.84615
C_all(N)	0	3.84615
C_all(P)	0	3.84615
C_all(Q)	11.76471	3.84615
C_all(R)	2.94118	0
C_all(S)	0	3.84615

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
C_all(T)	2.94118	11.53846
C_all(V)	2.94118	3.84615
C_all(W)	2.94118	3.84615
C_all(Y)	0	0
AI	143.52941	105
II	53.39706	17.81538
KLEP840101_CH	4	7
KLEP840101_avg_c	0.11765	0.26923
CHAM830107_CH	5	2
CHAM830107_avg_c	0.14706	0.07692
CHAM830108_CH	13	12
CHAM830108_avg_c	0.38235	0.46154
HOPT810101	0.3	0.01923
KUHL950101	0.90412	0.83231
KYTJ820101	-0.18235	-0.11923
CIDH920101	0.02	0.08538
CIDH920102	0.25441	0.345
CIDH920103	0.25118	0.23769
CIDH920104	0.13529	0.23577
CIDH920105	0.16588	0.24962
EISD840101	-0.17735	-0.13577
GOLD730101	1.49147	1.62615
JOND750101	1.56412	1.68038
MANP780101	13.14794	12.96346
PONP800101	12.48853	12.43462
PONP800104	13.01412	12.98538
PONP800105	14.06971	14.00423
PONP800106	11.64853	11.47846
PRAM900101	9.72059	5.46538
SWER830101	-0.02765	0.09038

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
ZIMJ680101	1.49647	1.56923
JURD980101	-0.285	-0.20192
WOLR790101	0.09971	0.18
KIDA850101	0.30647	0.22538
CASG920101	-0.24118	-0.16538
ENGD860101	2.32353	1.30769
FASG890101	-0.42294	-0.36923
TOSSI2002	-0.92647	-0.77692
C_hyR(HP)	0	3.84615
C_hyR(CFLMVWI)	38.23529	38.46154
C_hyR(NQSTY)	14.70588	23.07692
C_hyR(AG)	11.76471	7.69231
C_hyR(DEKR)	35.29412	26.92308
C_b50(P)	0	3.84615
C_b50(ST)	2.94118	15.38462
C_b50(CLVIM)	35.29412	30.76923
C_b50(AG)	11.76471	7.69231
C_b50(KR)	23.52941	26.92308
C_b50(H)	0	0
C_b50(FWY)	2.94118	7.69231
C_b50(DENQ)	23.52941	7.69231
C_cs(VIT)	23.52941	34.61538
C_cs(P)	0	3.84615
C_cs(A)	8.82353	3.84615
C_cs(SCMEQKRL)	61.76471	42.30769
C_cs(G)	2.94118	3.84615
C_cs(DN)	0	3.84615
C_cs(HFYW)	2.94118	7.69231
C_hydT(GASTPHY)	14.70588	26.92308
C_hydT(CLVIMFW)	38.23529	38.46154

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
C_hydT(RKEDQN)	47.05882	34.61538
C_vw(MHKFRYW)	26.47059	38.46154
C_vw(NVEQIL)	58.82353	34.61538
C_vw(GASTCPD)	14.70588	26.92308
C_pol(PATGS)	14.70588	26.92308
C_pol(HQRKNED)	47.05882	34.61538
C_pol(LIFWCMVY)	38.23529	38.46154
C_polz(KMHFRYW)	26.47059	38.46154
C_polz(GASDT)	14.70588	23.07692
C_polz(CPNVEQIL)	58.82353	38.46154
C_chrg(DE)	11.76471	0
C_chrg(ANCQGHILMFPSTWYV)	64.70588	73.07692
C_chrg(KR)	23.52941	26.92308
C_ss(EALMQKRH)	70.58824	42.30769
C_ss(VIYCWFT)	26.47059	42.30769
C_ss(GNPSD)	2.94118	15.38462
C_sa(ALFCGIVW)	50	42.30769
C_sa(MPSTHY)	2.94118	23.07692
C_sa(RKQEND)	47.05882	34.61538
T_hydT(GASTPHY->CLVIMFW)	18.18182	28
T_hydT(GASTPHY->RKEDQN)	12.12121	20
T_hydT(CLVIMFW->RKEDQN)	36.36364	24
T_vw(MHKFRYW->GASTCPD)	6.06061	24
T_vw(NVEQIL->GASTCPD)	24.24242	24
T_vw(MHKFRYW->NVEQIL)	30.30303	24
T_pol(PATGS->HQRKNED)	12.12121	20
T_pol(HQRKNED->LIFWCMVY)	36.36364	24
T_pol(PATGS->LIFWCMVY)	18.18182	28
T_polz(KMHFRYW->CPNVEQIL)	30.30303	28
T_polz(KMHFRYW->GASDT)	6.06061	20

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
T_polz(GASDT->CPNVEQIL)	24.24242	20
T_chrg(DE->KR)	6.06061	0
T_chrg(DE->ANCQGHILMFPSTWYV)	18.18182	0
T_chrg(ANCQGHILMFPSTWYV->KR)	24.24242	44
T_ss(EALMQKRH->GNPSD)	0	16
T_ss(VIYCWFT->GNPSD)	6.06061	16
T_ss(EALMQKRH->VIYCWFT)	33.33333	24
T_sa(ALFCGIVW->MPSTHY)	6.06061	24
T_sa(MPSTHY->RKQEND)	0	12
T_sa(ALFCGIVW->RKQEND)	48.48485	32
D_0_hydT(GASTPHY)	20.58824	7.69231
D_0_hydT(CLVIMFW)	2.94118	3.84615
D_0_hydT(RKEDQN)	5.88235	11.53846
D_25_hydT(GASTPHY)	20.58824	26.92308
D_25_hydT(CLVIMFW)	23.52941	23.07692
D_25_hydT(RKEDQN)	17.64706	15.38462
D_50_hydT(GASTPHY)	64.70588	38.46154
D_50_hydT(CLVIMFW)	61.76471	46.15385
D_50_hydT(RKEDQN)	35.29412	69.23077
D_75_hydT(GASTPHY)	73.52941	50
D_75_hydT(CLVIMFW)	76.47059	65.38462
D_75_hydT(RKEDQN)	58.82353	84.61538
D_100_hydT(GASTPHY)	91.17647	92.30769
D_100_hydT(CLVIMFW)	100	88.46154
D_100_hydT(RKEDQN)	94.11765	100
D_0_vw(MHKFRYW)	5.88235	3.84615
D_0_vw(NVEQIL)	2.94118	11.53846
D_0_vw(GASTCPD)	20.58824	7.69231
D_25_vw(MHKFRYW)	8.82353	42.30769
D_25_vw(NVEQIL)	23.52941	19.23077

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
D_25_vw(GASTCPD)	20.58824	26.92308
D_50_vw(MHKFRYW)	47.05882	57.69231
D_50_vw(NVEQIL)	52.94118	46.15385
D_50_vw(GASTCPD)	64.70588	38.46154
D_75_vw(MHKFRYW)	70.58824	88.46154
D_75_vw(NVEQIL)	76.47059	65.38462
D_75_vw(GASTCPD)	73.52941	50
D_100_vw(MHKFRYW)	94.11765	100
D_100_vw(NVEQIL)	100	80.76923
D_100_vw(GASTCPD)	91.17647	92.30769
D_0_pol(PATGS)	20.58824	7.69231
D_0_pol(HQRKNED)	5.88235	11.53846
D_0_pol(LIFWCMVY)	2.94118	3.84615
D_25_pol(PATGS)	20.58824	26.92308
D_25_pol(HQRKNED)	17.64706	15.38462
D_25_pol(LIFWCMVY)	23.52941	23.07692
D_50_pol(PATGS)	64.70588	38.46154
D_50_pol(HQRKNED)	35.29412	69.23077
D_50_pol(LIFWCMVY)	61.76471	46.15385
D_75_pol(PATGS)	73.52941	50
D_75_pol(HQRKNED)	58.82353	84.61538
D_75_pol(LIFWCMVY)	76.47059	65.38462
D_100_pol(PATGS)	91.17647	92.30769
D_100_pol(HQRKNED)	94.11765	100
D_100_pol(LIFWCMVY)	100	88.46154
D_0_polz(KMHFRYW)	5.88235	3.84615
D_0_polz(GASDT)	20.58824	7.69231
D_0_polz(CPNVEQIL)	2.94118	11.53846
D_25_polz(KMHFRYW)	8.82353	42.30769
D_25_polz(GASDT)	20.58824	26.92308

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
D_25_polz(CPNVEQIL)	23.52941	23.07692
D_50_polz(KMHFRYW)	47.05882	57.69231
D_50_polz(GASDT)	64.70588	30.76923
D_50_polz(CPNVEQIL)	52.94118	46.15385
D_75_polz(KMHFRYW)	70.58824	88.46154
D_75_polz(GASDT)	73.52941	73.07692
D_75_polz(CPNVEQIL)	76.47059	65.38462
D_100_polz(KMHFRYW)	94.11765	100
D_100_polz(GASDT)	91.17647	92.30769
D_100_polz(CPNVEQIL)	100	80.76923
D_0_ss(EALMQKRH)	5.88235	3.84615
D_0_ss(VIYCWFT)	2.94118	19.23077
D_0_ss(GNPSD)	73.52941	26.92308
D_25_ss(EALMQKRH)	26.47059	11.53846
D_25_ss(VIYCWFT)	14.70588	30.76923
D_25_ss(GNPSD)	0	26.92308
D_50_ss(EALMQKRH)	47.05882	46.15385
D_50_ss(VIYCWFT)	64.70588	61.53846
D_50_ss(GNPSD)	73.52941	38.46154
D_75_ss(EALMQKRH)	79.41176	69.23077
D_75_ss(VIYCWFT)	70.58824	73.07692
D_75_ss(GNPSD)	73.52941	50
D_100_ss(EALMQKRH)	100	100
D_100_ss(VIYCWFT)	97.05882	92.30769
D_100_ss(GNPSD)	73.52941	80.76923
D_0_chrg(DE)	11.76471	0
D_0_chrg(ANCQGHILMFPSTWYV)	2.94118	3.84615
D_0_chrg(KR)	5.88235	15.38462
D_25_chrg(DE)	11.76471	0
D_25_chrg(ANCQGHILMFPSTWYV)	41.17647	23.07692

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
D_25_chrg(KR)	8.82353	42.30769
D_50_chrg(DE)	17.64706	0
D_50_chrg(ANCQGHILMFPSTWYV)	61.76471	46.15385
D_50_chrg(KR)	29.41176	69.23077
D_75_chrg(DE)	32.35294	0
D_75_chrg(ANCQGHILMFPSTWYV)	82.35294	65.38462
D_75_chrg(KR)	50	84.61538
D_100_chrg(DE)	38.23529	0
D_100_chrg(ANCQGHILMFPSTWYV)	100	92.30769
D_100_chrg(KR)	94.11765	100
D_0_sa(ALFCGIVW)	2.94118	7.69231
D_0_sa(MPSTHY)	64.70588	3.84615
D_0_sa(RKQEND)	5.88235	11.53846
D_25_sa(ALFCGIVW)	23.52941	23.07692
D_25_sa(MPSTHY)	0	26.92308
D_25_sa(RKQEND)	17.64706	15.38462
D_50_sa(ALFCGIVW)	61.76471	46.15385
D_50_sa(MPSTHY)	64.70588	30.76923
D_50_sa(RKQEND)	35.29412	69.23077
D_75_sa(ALFCGIVW)	76.47059	61.53846
D_75_sa(MPSTHY)	64.70588	73.07692
D_75_sa(RKQEND)	58.82353	84.61538
D_100_sa(ALFCGIVW)	100	88.46154
D_100_sa(MPSTHY)	64.70588	92.30769
D_100_sa(RKQEND)	94.11765	100
[GASTPHY][RKEDQN]	3.0303	16
[CLVIMFW][GASTPHY]	6.06061	20
[RKEDQN][CLVIMFW]	15.15152	16
[GASTPHY][GASTPHY]	0	4
[RKEDQN][GASTPHY]	9.09091	4

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
[CLVIMFW][CLVIMFW]	9.09091	12
[GASTPHY][CLVIMFW]	12.12121	8
[RKEDQN][RKEDQN]	24.24242	12
[CLVIMFW][RKEDQN]	21.21212	8
[GASTPHY][CLVIMFW][RKEDQN]	9.375	4.16667
[GASTPHY][RKEDQN][CLVIMFW]	3.125	8.33333
[CLVIMFW][RKEDQN][CLVIMFW]	3.125	0
[RKEDQN][CLVIMFW][GASTPHY]	3.125	8.33333
[RKEDQN][GASTPHY][RKEDQN]	3.125	0
[CLVIMFW][CLVIMFW][GASTPHY]	3.125	4.16667
[RKEDQN][RKEDQN][CLVIMFW]	9.375	8.33333
[CLVIMFW][CLVIMFW][RKEDQN]	3.125	4.16667
[GASTPHY][GASTPHY][CLVIMFW]	0	4.16667
[CLVIMFW][GASTPHY][CLVIMFW]	6.25	0
[GASTPHY][CLVIMFW][GASTPHY]	0	4.16667
[GASTPHY][RKEDQN][RKEDQN]	0	8.33333
[CLVIMFW][RKEDQN][RKEDQN]	12.5	4.16667
[RKEDQN][GASTPHY][GASTPHY]	0	0
[RKEDQN][CLVIMFW][RKEDQN]	6.25	0
[GASTPHY][GASTPHY][RKEDQN]	0	0
[RKEDQN][GASTPHY][CLVIMFW]	6.25	4.16667
[CLVIMFW][GASTPHY][RKEDQN]	0	16.66667
[GASTPHY][CLVIMFW][CLVIMFW]	3.125	0
[RKEDQN][RKEDQN][RKEDQN]	12.5	0
[GASTPHY][RKEDQN][GASTPHY]	0	0
[CLVIMFW][CLVIMFW][CLVIMFW]	0	4.16667
[RKEDQN][CLVIMFW][CLVIMFW]	6.25	8.33333
[CLVIMFW][RKEDQN][GASTPHY]	6.25	4.16667
[RKEDQN][RKEDQN][GASTPHY]	3.125	0
[GASTPHY][GASTPHY][GASTPHY]	0	0

Continúa en la página siguiente

Tabla 22 – continúa de la página anterior

ID	DRAMP16957	DBAASP5756
[CLVIMFW][GASTPHY][GASTPHY]	0	4.16667
length	34	26
mw	4001.89895	3015.78085
Z(pH_5)	4.48529	7.03802
Z(pH_7)	3.97981	6.97479
Z(pH_9)	3.17527	6.17554
boman	1.13324	0.68538
hydro	0.06636	0.04909
HM(100)	0.40093	0.47989
HM(160)	0.36086	0.21001
HM(180)	0.28636	0.28182
pI	10.67349	11.57872

A.2. Hiperparámetros de los algoritmos de aprendizaje de máquina

En esta sección se presentan los hiperparámetros que se utilizaron para los algoritmos de aprendizaje de máquina RF y SVM-L para los conjuntos de datos DAT1, DAT2 y DAT3, respectivamente. Los hiperparámetros se muestran tal y como se asignaron en la librería para Java Weka 3.8.

Los hiperparámetros para el algoritmo de aprendizaje máquina RF son los siguientes:

```
Scheme:
weka.classifiers.functions.LibLINEAR -S 1 -C 0.01 -E 0.001 -B 1.0 -L 0.1 -I 1000
```

Por otra parte, los hiperparámetros para el algoritmo de aprendizaje máquina SVM-L para el conjunto de datos DAT2 y DAT3 son los siguientes:

```
Scheme:
weka.classifiers.functions.LibLINEAR -S 1 -C 1.0 -E 0.001 -B 1.0 -L 0.1 -I 1000
```

Hiperparámetros para el algoritmo de aprendizaje máquina SVM-L para el conjunto de datos DAT1:

```
Scheme:
weka.classifiers.functions.LibLINEAR -S 1 -C 0.01 -E 0.001 -B 1.0 -L 0.1 -I 1000
```

A.3. Evaluación del modelo generativo para el diseño de AMPs

En esta sección se muestra el desempeño que obtuvo el modelo generativo bidireccional LSTM en el entrenamiento y validación. Adicionalmente comparamos el desempeño del modelo bidireccional LSTM con un modelo de una sola dirección LSTM (ver Figura 40).

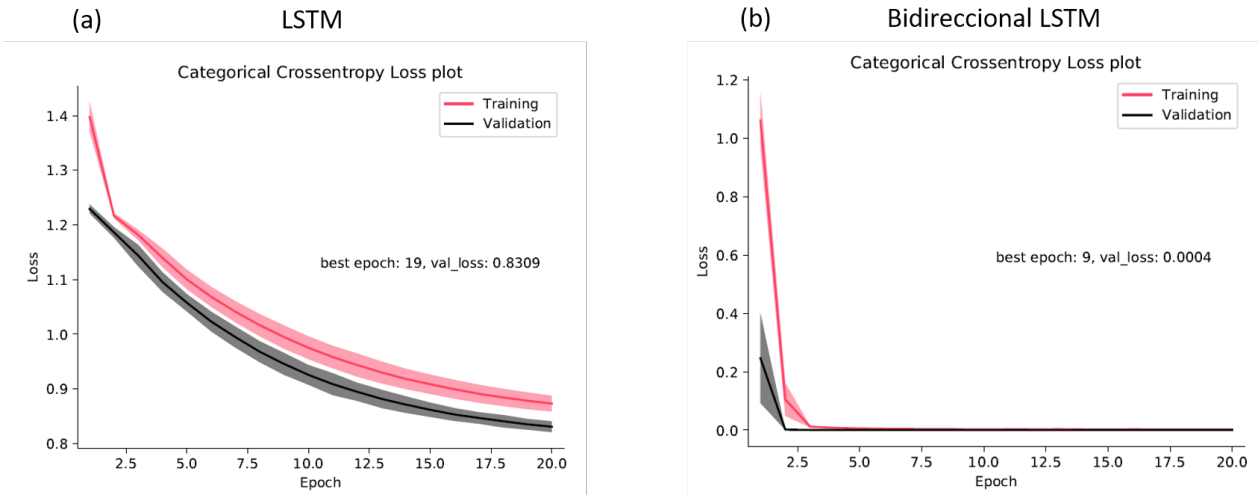


Figura 40. El promedio y la desviación estándar de los parámetros críticos. Función de costo dentro de una validación cruzada de 5 pliegues (un modelo con una menor pérdida de costo es mejor). El promedio de la función de costo (línea sólida) y la desviación estándar (áreas sombreadas) para el conjunto de entrenamiento y la validación. (a) Comportamiento del modelo generativo de memoria a largo y corto plazo (LSTM). (b) Modelo generativo bidireccional de memoria a largo y corto plazo, la cual procesa las secuencias en ambas direcciones conectando las capas ocultas de direcciones opuestas a la misma salida.