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

Maestría en Ciencias en Ciencias de la Computación

Implementación de algoritmos de clasificación de una sola clase para la clasificación de péptidos antimicrobianos

Tesis

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

Presenta:

Isaac Pedro Tapia Contreras

Ensenada, Baja California, México

2022

Tesis defendida por

Isaac Pedro Tapia Contreras

y aprobada por el siguiente Comité

Dr. Carlos Alberto Brizuela Rodríguez

Codirector de tesis

Dr. César Raúl García Jacas

Codirector de tesis

Miembros del comité

Dr. Hugo Homero Hidalgo Silva

Dra. Patricia Juárez Camacho

Dr. Pedro Gilberto López Mariscal

Coordinador del Posgrado en Ciencias de la Computación

Dr. Pedro Negrete Regagnon

Director de Estudios de Posgrado

Resumen de la tesis que presenta Isaac Pedro Tapia Contreras como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Implementación de algoritmos de clasificación de una sola clase para la clasificación de péptidos antimicrobianos

Resumen aprobado por:

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

Desde el descubrimiento de péptidos con propiedades antimicrobianas, se ha generado un interés por encontrar nuevas secuencias que posean potencial terapéutico en la inhibición de agentes patógenos como: bacterias, hongos, virus, parásitos, entre otros. Para el descubrimiento de nuevas secuencias, se han utilizado de manera exitosa, una variedad de modelos de aprendizaje máquina basados en algoritmos de clasificación binaria y clasificación multiclase, como pueden ser: la máquina de soporte vectorial, bosque aleatorio, K-vecinos más cercanos, redes neuronales, entre otros. Una característica importante de estos algoritmos de clasificación, es que dependen de ejemplos tanto de la clase positiva (AMP), como de la negativa (No-AMP), para poder realizar su proceso de entrenamiento. El problema encontrado con esta metodología, es que actualmente, no se dispone de un conjunto de péptidos validados experimentalmente como no-antimicrobianos. Los conjuntos utilizados en la literatura, se obtienen recuperando secuencias que pertenezcan a organelos celulares (mitocondria, retículo endoplasmático, aparato de Golgi, etc.), lo cual no garantiza la obtención de péptidos no antimicrobianos. Por lo tanto, todos los modelos encontrados en la literatura que utilizan esta metodología, están entrenados utilizando un conjunto de entrenamiento negativo sesgado, que bien podría contener péptidos antimicrobianos desconocidos. Para atacar este problema, se propone utilizar una metodología de clasificación de una sola clase; esto debido a que los algoritmos de una clase, requieren únicamente de la clase positiva para realizar su entrenamiento, que en este caso, es la única que contiene ejemplos validados experimentalmente. En el presente trabajo, se propone un esquema de clasificación jerárquica de una clase, para discriminar “in silico”, secuencias de péptidos antibacteriales del tipo anti Gram positivo y anti Gram negativo. Se compara además el desempeño del esquema propuesto con uno tradicional. Resultados de los experimentos computacionales muestran que: i) los modelos jerárquicos presentan valores superiores de especificidad y los no jerárquicos de sensibilidad, ii) los mejores descriptores para esta tarea de discriminar anti Gram positivo de anti Gram negativo son los del tipo físico-químicos calculados por el software ProtDcal, superando incluso a los generados por la red neuronal profunda, BERT ESM.

Palabras clave: péptidos antimicrobianos, aprendizaje máquina, clasificación de una clase, péptidos Gram negativos, péptidos Gram positivos

Abstract of the thesis presented by Isaac Pedro Tapia Contreras as a partial requirement to obtain the Master of Science degree in Computer Science.

Implementation of one-class classification algorithms for antimicrobial peptide classification

Abstract approved by:

Dr. Carlos Alberto Brizuela Rodríguez
Thesis Co-Director

Dr. César Raúl García Jacas
Thesis Co-Director

Since the discovery of peptides with antimicrobial properties, finding new sequences with therapeutic potential in the inhibition of pathogens such as bacteria, fungi, viruses, parasites, among others has received increased interest. To discover new sequences, a variety of machine learning models based on binary classification and multiclass classification algorithms have been successfully used, some examples are: support vector machine, random forest, K-nearest neighbors, and neural networks. An important characteristic of these classification algorithms is that they depend on examples of both, the positive class (AMP) and the negative class (No-AMP) in order to perform their training process. The problem encountered with this methodology is that there is not a set of experimentally validated non-antimicrobial peptides available. The sets used in the literature are obtained by recovering sequences found in cellular organelles (mitochondria, endoplasmic reticulum, Golgi apparatus, etc.), which does not guarantee the absence of antimicrobial activities. Therefore, all models found in the literature based on this methodology are trained by using a biased negative training set, which may well contain unknown antimicrobial peptides. To address this problem, we propose to use a one-class classification methodology; this is because one-class algorithms require only the positive class to perform their training, which in this case, is the only one that contains experimentally validated examples. In the present work, a one-class hierarchical classification scheme is proposed to distinguish “in silico”, antibacterial peptides sequences of the anti Gram-positive and anti Gram-negative types. Additionally, the performance of the proposed scheme is compared with a traditional one-class approach. Results from the computational experiments show that: i) the hierarchical models present superior values of specificity and the non-hierarchical ones of sensitivity, ii) the best descriptors for this task of discriminating anti Gram positive from anti Gram negative sequences are those of the physico-chemical type calculated by the ProtDcal software, surpassing even those generated by the deep neural network, BERT ESM.

Keywords: antimicrobial peptides, machine learning, one-class classification, Gram-negative peptides, Gram-positive peptides, Gram-positive peptides

Dedicatoria

A mis padres, por su apoyo incondicional en cada etapa de mi desarrollo.

Agradecimientos

A mis directores de tesis, el Dr. Carlos Alberto Brizuela Rodríguez y el Dr. César Raúl García Jacas. Por darme la oportunidad de aprender de ellos, por guiarme y acompañarme desde el primer experimento hasta el último; y por su interminable paciencia y dedicación, que me permitió mejorar la calidad de mi trabajo.

A los miembros de mi comité de tesis, el Dr. Hugo Homero Hidalgo Silva y la Dra. Patricia Juárez Camacho. Por aceptar ser parte de este comité y por siempre hacer críticas constructivas a este trabajo.

A todo el personal del departamento de ciencias de la computación, por mantener una impecable calidad en el posgrado.

A todos mis compañeros de generación, principalmente a Maria Concepción Valdez y Misael Astorga, por su amistad y apoyo emocional a lo largo de esta tesis.

Al Centro de Investigación Científica y de Educación Superior de Ensenada por la oportunidad de realizar mis estudios de posgrado.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de maestría/doctorado. No. de becario: 961781.

Tabla de contenido

Página

Tabla de contenido (continuación)

Tabla de contenido (continuación)

Lista de figuras

Figura Página

Lista de figuras (continuación)

Figura Página

Lista de figuras (continuación)

Figura Página

Lista de tablas

Tabla Página

Lista de tablas (continuación)

Lista de tablas (continuación)

Lista de tablas (continuación)

Tabla Página

Lista de tablas (continuación)

Capítulo 1. Introducción

1.1. Antecedentes

Los péptidos son moléculas naturalmente sintetizadas tanto por células procariotas como eucariotas, e intervienen en una gran variedad de procesos celulares. Dentro de esta variedad de funciones se ha encontrado que existen péptidos que poseen la capacidad de combatir agentes patógenos como pueden ser bacterias, hongos, virus, parásitos, entre otros. A esta variedad de péptidos se le ha denominado como "péptidos antimicrobianos" (AMP por sus siglas en inglés "Antimicrobial Peptide") y desde su descubrimiento como parte del sistema inmune de varios organismos, se ha iniciado una carrera por encontrar secuencias que presenten estas propiedades (Andersson *et al.* (2016)).

Estas moléculas son encontradas en el sistema inmune innato de una variedad de organismos, como principalmente pueden ser: plantas, vertebrados, insectos, anfibios, entre otros (Zasloff, 2002). No existe una estructura peptídica o propiedad fisicoquímica que se presente en todos los péptidos antimicrobianos, sin embargo, existen características que son comunes en este tipo de péptidos (SJ *et al.* (2012)): i) miden entre 2 y 100 aminoácidos. Analizando la base de datos starPep (Aguilera-Mendoza *et al.* (2019)), se encuentra que más del 80 % de los péptidos antimicrobianos está en un intervalo de entre 5 y 50 aminoácidos de longitud; ii) tienden a tener una carga neta positiva, en un intervalo de +2 y +9, siendo más común aquellos que se encuentran en el intervalo entre +4 y +6. Gracias a su carga positiva es como los péptidos antimicrobianos pueden interactuar con la bicapa fosfolipídica de las bacterias, dado que esta posee una carga neta negativa.

Los péptidos antimicrobianos (AMP) se descubrieron en 1939, cuando se aislaron sustancias antimicrobianas, denominadas gramicidinas, a partir del organismo "Bacillus brevis", se descubrió que estas sustancias tenían actividad inhibidora de una gran variedad de bacterias Gram positivas (Dubos (1939)). Las gramicidinas fueron también, los primeros péptidos antimicrobianos utilizados como antibióticos de manera comercial y aunque su uso intravenoso está contra indicado, han tenido mucho éxito

siendo utilizadas en conjunto con antibióticos tópicos para el tratamiento de enfermedades cutáneas (Epps, 2006).

Aunque la investigación sobre los péptidos antimicrobianos continuó a lo largo del siglo XX, no fue hasta principios del siglo XXI, cuando la madurez de los algoritmos de aprendizaje máquina, permitió que se construyeran los primeros modelos computacionales para la clasificación de péptidos antimicrobianos, de la mano de la metodología QSAR (Richon y Young, 1997). La relación cuantitativa estructura-actividad (QSAR por sus siglas en inglés), es una metodología que propone que se pueden construir modelos estadísticos (o en su defecto de aprendizaje de máquina), que sean capaces de modelar una actividad biológica, utilizando únicamente, la información de moléculas con dicha actividad (Roy *et al.* (2015)).

La aparición de modelos QSAR modernos, fue posible en parte, gracias a la generación de bases de datos donde se compilan una gran cantidad de moléculas junto con la información de su estructura, propiedades fisicoquímicas, actividad biológica, etc. Específicamente para el caso de los péptidos antimicrobianos se han generado bases de datos que contienen secuencias de péptidos que han sido validadas experimentalmente, algunas de las que pueden ser encontradas en la literatura son: ADAM (Torrent *et al.* (2012)), The antimicrobial peptide database (APD3) (Wang *et al.* (2016)), Database of Antimicrobial Activity and Structure of Peptides (DBAASP) Pirtskhalava *et al.* (2016), Data Repository of Antimicrobial Peptides (DRAMP) (Kang *et al.* (2019)), Yet another database of antimicrobial peptides (YADAMP) (Piotto *et al.* (2012)), DbAMP (Jhong *et al.* (2019)).

También existen bases de datos que se enfocan en indexar péptidos que contengan una actividad específica, algunos ejemplos son: PlantAFP (Tyagi *et al.* (2019)), la cual contiene péptidos antifúngicos obtenidos en plantas; AVPdb (Qureshi *et al.* (2014)), la cual contiene péptidos validados con actividad antiviral; CancerPPD (Tyagi *et al.* (2015)). la cual contiene péptidos con actividad anticancerígena, ParaPep (Mehta *et al.* (2014)), la cual contiene péptidos antiparasitarios.

Utilizando las secuencias de las bases de datos de la literatura se han generado una variedad de modelos para la clasificación de péptidos antimicrobianos. La clasificación de péptidos antimicrobianos consiste en asignar correctamente una etiqueta a

una secuencia de péptido, que corresponde a la actividad biológica que este presenta (Chou, 2001).

Algunos ejemplos de la literatura donde se generan modelos de clasificación de péptidos antimicrobianos son: el modelo presentado por Lata *et al.* (2007), este es uno de los primeros modelos encontrados en la literatura, en este se utilizó un algoritmo de máquina de soporte vectorial (Evgeniou y Pontil, 2001) para generar el modelo, el cual obtiene un 91.66 % de precisión en su validación; el modelo presentado por Xiao *et al.* (2013), que utiliza el algoritmo de K vecinos más cercanos (Fix y Hodges (1989)) para generar el modelo de clasificación y que logra un 92.23 % de precisión utilizando esta metodología y el modelo de Lin *et al.* (2019), que modela 16 clases de péptidos antimicrobianos utilizando el algoritmo bosque aleatorio (Breiman (2001)) y obtiene en promedio una precisión del 85.5 %. Los modelos anteriormente presentados, son generados utilizando técnicas tradicionales de aprendizaje máquina. Sin embargo, existen también ejemplos en la literatura, como el de Veltri *et al.* (2018) y el de Pinacho-Castellanos *et al.* (2021b), que utilizan técnicas de aprendizaje profundo para generar sus modelos y que obtienen resultados similares o superiores a los de los métodos tradicionales.

En la presente investigación se propone utilizar algoritmos de clasificación de una sola clase, sustituyendo a los algoritmos binarios, tradicionalmente utilizados en la literatura. La clasificación de una sola clase se ha utilizado principalmente en problemas donde se tienen conjuntos de datos muy desbalanceados o donde obtener información de una de las clases es prácticamente imposible (Alam *et al.* (2020)). Aunque los clasificadores de una sola clase no han sido utilizados para la clasificación de péptidos antimicrobianos, sí se han utilizado exitosamente para otras aplicaciones QSAR, como es el caso de reciente de Vriza *et al.* (2021), quienes utilizaron algoritmos de una clase para generar modelos que asistieran en el descubrimiento de co-cristales de hidrocarburos poli aromáticos. La razón por la cual eligieron utilizar modelos de una clase en este caso es por la dificultad de encontrar muestras negativas.

1.2. Motivación

La principal motivación para implementar algoritmos de clasificación de una sola clase, para la clasificación de péptidos antimicrobianos es que estos requieren de ejemplos únicamente de la clase “objetivo” (positiva) para ser entrenados. Esta cualidad es deseable dado que existe una dificultad para obtener conjuntos negativos experimentalmente validados. Es decir, conjuntos negativos para los cuales se comprobó experimentalmente (mediante pruebas de laboratorio) que no poseen actividad antimicrobiana alguna.

Idealmente, es posible obtener un conjunto de péptidos experimentalmente validados como no antimicrobianos, para esto se requiere probar cada péptido candidato para todas las actividades conocidas que se consideren antimicrobianos, lo cual resulta en un trabajo bastante arduo y costoso. Para evitar esto, Xiao *et al.* (2013) y Veltri *et al.* (2018) proponen tomar las secuencias de las bases de datos públicas de péptidos y proteínas como pueden ser PubMed o Uniprot utilizando criterios de búsqueda que recupere secuencias con una baja probabilidad de poseer propiedades antimicrobianas. Los criterios de búsqueda pueden incluir conceptos como los siguientes: “no antimicrobiano”, “no antibacterial”, “no antibiótica”, “no citoplasma” y “no excretorias”.

Al usar la metodología propuesta en Xiao *et al.* (2013) y Veltri *et al.* (2018) para obtener los conjuntos negativos de entrenamiento y prueba, se están generando conjuntos en los cuales es posible se encuentren secuencias con actividad antimicrobiana. Por lo tanto, los modelos donde se utilicen estas secuencias, en su proceso de entrenamiento, estarán diseñados para rechazar todas las secuencias pertenecientes o similares a las que conforman estos conjuntos. Esto nos lleva a identificar una oportunidad de mejora: si utilizamos modelos de una clase para el proceso de clasificación, estaríamos entrenando un modelo sin este sesgo y que podría ser capaz de reconocer péptidos antimicrobianos del conjunto negativo, en el caso que estos existan.

1.3. Objetivos

1.3.1. Objetivo general

Evaluar el desempeño de los algoritmos de clasificación de una sola clase cuando son utilizados en la clasificación de péptidos antimicrobianos.

1.3.2. Objetivos específicos

- Definir conjuntos de entrenamiento y de prueba para ser utilizados en la implementación de algoritmos de clasificación de una sola clase.
- Proponer una metodología de clasificación de péptidos antimicrobianos utilizando algoritmos de clasificación de una sola clase encontrados en la literatura.
- Determinar una jerarquía de los clasificadores implementados en el objetivo anterior, en función del desempeño relativo de estos.

1.4. Metodología propuesta

Con el fin de cumplir con los objetivos planteados, se propuso la siguiente metodología. Primero, se construyeron los conjuntos de entrenamiento y validación, utilizando la base de datos starPep para la obtención de secuencias. Durante este proceso, las muestras se seleccionaron con base en su actividad biológica y longitud. Después, estas secuencias se sometieron a un proceso de descripción molecular. Posteriormente, se sometieron los descriptores moleculares a un proceso de selección de características. Una vez se obtuvo el conjunto final de descriptores, estos se utilizaron en un proceso de clusterización de los conjuntos de entrenamiento. Después, la representación completa de los conjuntos de entrenamiento y la información de los clusters, se utilizaron para generar un modelo jerárquico de clasificación. Por último, se midió el desempeño de los modelos generados y se comparó el desempeño de estos con modelos binarios reportados en la literatura.

1.5. Organización de la tesis

Toda la información obtenida para la elaboración de esta tesis se concentró en 5 capítulos y dos anexos.

En el primer capítulo, se presenta una breve introducción a los péptidos antimicrobianos. Se presentan también algunos de los modelos de aprendizaje de máquina que se utilizan en la literatura para clasificar péptidos antimicrobianos, y la motivación que nos llevó a la utilización de los algoritmos de una sola clase.

En el segundo capítulo, presentamos una generalización de la metodología QSAR, así como los métodos, aplicaciones y recursos que comúnmente son utilizados en la generación de clasificadores de péptidos antimicrobianos. También se presenta una introducción al funcionamiento de los algoritmos de una sola clase, utilizando como ejemplos los algoritmos Local Outlier Factor e Isolation Forest.

En el tercer capítulo, se describe la metodología que se desarrolló utilizando clasificadores de una sola clase. Se presenta la metodología de obtención de conjuntos de entrenamiento y validación, así como los procesos de descripción molecular, entrenamiento de modelos y clasificación jerárquica que se utilizaron a lo largo de la experimentación.

En el cuarto capítulo, se presentan los resultados obtenidos a partir de la metodología desarrollada en el capítulo tres. En este capítulo se incluyen todos los experimentos realizados así como una comparación de los modelos obtenidos con los del estado del arte.

En el quinto capítulo, se presenta una discusión sobre los resultados obtenidos, haciendo énfasis en su desempeño y los factores que afectan al mismo. Así mismo, se presentan las conclusiones que se derivan de los resultados obtenidos en el cuarto capítulo. Por último, se presentan como trabajo futuro un conjunto de oportunidades de investigación que quedaron fuera del alcance de esta tesis.

Se incluyen también dos anexos, el Anexo A, contiene la información de todos los conjuntos de secuencias utilizados, y en el Anexo B se presentan todos los resultados obtenidos durante la experimentación.

Capítulo 2. Marco Teórico

En este capítulo se incluyen conocimientos básicos necesarios para poder entender y replicar la metodología presentada en Capítulo 3. Principalmente, se expone estos puntos a través de la metodología QSAR.

2.1. Metodología QSAR

El modelado QSAR o “Relación Cuantitativa Estructura-Actividad (QSAR, Quantitative Structure-Activity Relationship) busca el desarrollo de una correlación matemática entre una respuesta química y los atributos químicos cuantitativos que definen las características de las moléculas analizadas. Por lo tanto, esta metodología intenta establecer un formalismo matemático (Función matemática) entre el comportamiento/actividad de una sustancia química, es decir, entre la respuesta química y un conjunto de atributos químicos cuantitativos que pueden extraerse de las estructuras químicas utilizando medios experimentales o teóricos (Roy *et al.* (2015)).

Los modelos QSAR son generados con la suposición de que tanto la estructura como las propiedades fisicoquímicas de una molécula deben tener una relación directa en sus propiedades químicas, físicas o biológicas. Por lo tanto, se infiere que la actividad biológica o reactividad de una sustancia química se puede predecir a partir de la información de moléculas similares cuyas actividades ya han sido evaluadas y que son positivas para la actividad o “endpoint” que se desea modelar (Todeschini y Consonni (2009)). La definición básica del modelado QSAR puede ser representado por la siguiente ecuación:

$$Actividad\ Biolgica = f(\text{Propiedades fisicoquímicas, información estructural}) \quad (1)$$

tomando en cuenta la definición de un modelo QSAR como un modelo matemático y considerando que este modelo proviene de tomar la contribución individual de las distintas propiedades de las moléculas explicadas en la ecuación 1. Podemos definir de manera matemática un modelo QSAR como:

$$Y = F(X_1, X_2, X_3, X_4, X_5, \ldots, X_n) \quad (2)$$

donde la variable Y representa la actividad biológica que se está modelando mientras las variables X representan cada una de las propiedades estructurales o fisicoquímicas que se están incluyendo en el modelo. A partir de la representación de un modelo QSAR encontrada en la ecuación 2, es que podemos empezar a intuir como es que esta metodología extrapola la información de moléculas conocidas a nuevas moléculas (Roy *et al.* (2015)).

A partir de las ecuaciones 1 y 2, podemos intuir que para la generación de un modelo QSAR se requiere de al menos los siguientes tres componentes: I) un conjunto de moléculas que posean la actividad biológica que se desea modelar; II) una forma de representar las moléculas analizadas de forma numérica; III) un modelo matemático capaz de correlacionar la representación numérica de las moléculas con la actividad biológica que se está modelando (Roy *et al.* (2015)). En la práctica, el conjunto de moléculas puede ser obtenido de una de las bases de datos que existen en la literatura para diversos tipos de moléculas y actividades biológicas; la representación numérica puede ser obtenida utilizando descriptores moleculares, y como modelo matemático se pueden utilizar modelos estadísticos, de aprendizaje máquina o aprendizaje profundo (Todeschini y Consonni (2009)).

La metodología QSAR se puede considerar como una plantilla para la generación de un modelo de una actividad biológica. No existe como tal, un conjunto de descriptores o algoritmos que sea universalmente aceptados para todos los tipos de moléculas y actividades biológicas. Sin embargo, se han realizado por parte de muchos autores, una serie de sugerencias que pueden seguirse para la generación de un modelo QSAR robusto. En Nantasenamat (2020) se sugieren una serie de pasos para generar un modelo QSAR robusto y reproducible:

- **Compilación de datos:** durante este proceso se recaba la información de las moléculas que se desea investigar. Las fuentes de datos potenciales incluyen la literatura y bases de datos de actividad biológica pre-procesadas.
- **Pre-procesamiento de la información:** En esta etapa se limpia la información

de todas las características indeseables que esta pueda contener; como información incompleta, redundante o que pueda afectar el desempeño del modelo.

- **Separación/balanceo de datos:** una práctica común en la generación de modelos de aprendizaje máquina, es la de separar el conjunto de datos obtenidos en dos conjuntos, uno de entrenamiento y otro de validación. Existen varias técnicas para llevar a cabo esta separación, siendo la más común, la partición aleatoria del conjunto.

- **Cálculo de descriptores moleculares:** En esta etapa se seleccionan el conjunto de propiedades moleculares que se utilizará para definir el modelo, estas deben ser congruentes con el tipo de molécula que se está analizando y la actividad biológica que se desea modelar.

- **Selección de características:** Una vez generados los descriptores, el conjunto inicial de descriptores se somete normalmente a la eliminación de las variables de baja variabilidad, seguida de la eliminación de las variables redundantes.

- **Modelo matemático/ modelo de aprendizaje:** El punto culminante del proceso de construcción del modelo QSAR es el uso de los datos antes mencionados para el análisis multivariable con el fin de correlacionar los descriptores calculados con la actividad biológica modelada.

- **Medición de robustez / validación:** Un proceso común en el aprendizaje de máquina, es la etapa de validación. Aquí sé prueba la capacidad del modelo de realizar predicciones acertadas sobre los conjuntos de prueba.

El principal atractivo de la metodología QSAR es la flexibilidad que presenta en la modelación de actividades biológicas. Existen en la literatura cientos de bases de datos, descriptores moleculares y modelos de aprendizaje que han abierto el panorama a una nueva generación de descubrimiento de fármacos e investigación en química informática, bioinformática, etc.

2.2. Descriptores Moleculares

Un descriptor molecular se define como: “valores numéricos asociados a la constitución química para la correlación de la estructura química con varias propiedades físicas, actividad química o actividad biológica” (Roy *et al.* (2015)). De esta definición se obtiene que los descriptores moleculares son mediciones de una característica molecular y que a su vez esta medición puede ser representada con una magnitud escalar.

Dado que en el modelado QSAR se cuenta únicamente con los descriptores para representar a las moléculas estudiadas, es deseable encontrar aquellos descriptores que beneficien a la exactitud del modelo construido con estos. Según Roy *et al.* (2015) los descriptores moleculares utilizados en un modelo QSAR deben contar con las siguientes características:

- Un descriptor debe ser relevante (aplicable) para una amplia clase de compuestos
- Un descriptor debe estar correlacionado con las actividades biológicas estudiadas y demostrar al mismo tiempo una correlación insignificante con otros descriptores.
- El cálculo del descriptor debe ser rápido e independiente de las propiedades experimentales.
- Un descriptor debe producir valores diferentes para moléculas estructuralmente diferentes, aunque las diferencias estructurales sean pequeñas.
- Un descriptor debe poseer una capacidad de interpretación física para determinar las características del conjunto de los compuestos estudiados.

2.3. Descriptores moleculares para secuencias de péptidos y proteínas

Existen una variedad de métodos para codificar proteínas con la finalidad de generar modelos de aprendizaje de máquina. Aquí se presenta una serie de software para la generación de descriptores moleculares para péptidos y proteínas.

Tabla 1. Características utilizables en starPep para generar descriptores moleculares. Se enlistan las propiedades fisicoquímicas, grupos de aminoácidos y operadores de agregación que el software starPep puede utilizar para generar un descriptor molecular. Para generar un descriptor es necesario seleccionar al menos uno de cada grupo.

Propiedades fisicoquímicas	Grupos de aminoácidos	Operadores de agregación
Relative reverse-turn Frequency	Aliphatic Residues	Manhattan Norm
Geometric compatibility	Favoring Alpha Helix Residues	Euclidean Norm
Heat of formation	Apolar Residues	Arithmetic Mean
Side chain mass	Aromatic Residues	Quadratic Mean
Side chain volume	Favoring Beta Sheet Residues	Potential Mean
Isoelectric point	Favoring Beta Turn Residues	Harmonic Mean
Relative alpha-helix frequency	Negatively Charged Polar Residues	Variance
Relative beta-sheet frequency	Positively Charged Polar Residues	Skewness
Isotropic Surface area	Uncharged Polar Residues	Kurtosis
Z1-scale	Unfolding Residues	Standard Deviation
Z2-scale		Variation Coefficient
Z3-scale		Range
Boman		Inter-Percentile difference
Charge		
Hydrophilicity		

2.3.1. Descriptores de starPep

El software starPep (Aguilera-Mendoza *et al.* (2019)) contiene una herramienta capaz de generar descriptores moleculares basados en propiedades fisicoquímicas, cuenta además con diferentes operadores de agregación. Los operadores de agregación son funciones matemáticas que se utilizan para combinar información (Calvo *et al.* (2002)); esta función toma como entrada un conjunto de valores numéricos y obtiene un valor único que representa a los valores de entrada en alguna magnitud, algunos ejemplos de operadores de agregación comunes y que son utilizados por starPep son: media aritmética, desviación estándar, media harmónica, entre otros. En la Tabla 1 se muestran las posibles opciones de índices de propiedades, grupos químicos y operadores de agregación.

2.3.2. ProtDcal

Los descriptores moleculares de ProtDcal (Romero-Molina *et al.* (2019)) se generaron con la intención de describir la estabilidad del plegado de las proteínas y los factores que contribuyen a ella, es decir, la entropía configuracional, las interacciones

Tabla 2. Características utilizables de ProtDcal para generar descriptores moleculares. Se enlistan las propiedades fisicoquímicas, grupos de aminoácidos y operadores de agregación que el software ProtDcal puede utilizar para generar un descriptor molecular. Para generar un descriptor es necesario seleccionar al menos uno de cada grupo.

Propiedades fisicoquímicas	Grupos de aminoácidos	Operadores de agregación
Molecular weight	Alfa Helix Structure Residues	Manhattan Norm
Hydrophobicity	Beta Sheet Structure Residues	Euclidean Norm
Probability of alpha helix	Reverse Turn Structure Residues	Arithmetic Mean
Polar area of each aa unfolded	Positively Charged Residues	Harmonic Mean
Isoelectric point	Negatively Charged Residues	Potencial Mean
Probability of being beta sheet	Uncharged Residues	Geometric Mean
Electronic Charge index	Aromatic Residues	Potential mean
Isotropic Surface Area	Aliphatic Residues	Kurtosis
Probability of being beta turn	Unfolding Residues	Range
Z1	Non-Polar Residues	Variance
Z2	Polar Residues	Variation Coefficient
Z3	Loop regions residues	Standard Deviation
Heat of formation	Internal Residues	Minimum Value
Torsional compatibility	Superficial Residues	Maximum Value
Distance compatibility		Percentile 25 Q1
		Percentile 50 Q2
		Percentile 75 Q2
		Q3-Q1 I50

de empacamiento cercano y el efecto hidrofóbico. De manera similar a starPep, ProtDcal combina las diferentes propiedades fisicoquímicas y estructurales con operadores de agregación para generar descriptores moleculares. En la Tabla 2 se muestran las posibles opciones de índices de propiedades, grupos químicos y operadores de agregación.

2.3.2.1. Cálculo de descriptores de starPep y ProtDcal

Los descriptores moleculares de starPep y ProtDcal se calculan utilizando la siguiente metodología: primero se obtienen los valores de referencia de las propiedades fisicoquímicas (masa de la cadena lateral, índice calórico, punto isoeléctrico, etc.) de cada uno de los aminoácidos que forman la cadena del péptido al que se le calculan sus descriptores moleculares. Después se aplican los operadores de agregación sobre el total del conjunto de valores obtenidos en el paso anterior, así como también se calculan para grupos específicos de aminoácidos (negativamente cargados, positivamente cargados, aquellos que favorecen la formación de hélice alfa, etc.). El resultado

final es un conjunto de valores numéricos (descriptores moleculares) que codifican información sobre la estructura y función de las secuencias que se están estudiando. En la Figura 1 se muestra un diagrama explicando cómo calcular descriptores de starPep y ProtDcal basados en el coeficiente hidrófilo para una secuencia de la base de datos de starPep. A continuación se presentan los pasos realizados por starPep y ProtDcal para calcular un descriptor molecular:

1. Se obtiene el valor de referencia de cada propiedad fisicoquímica seleccionada para cada uno de los aminoácidos de la cadena que se está describiendo.
2. Se agrupan los valores de referencia en diferentes conjuntos de “fragmentos químicos”
3. Se aplican los operadores de agregación clásicos para el conjunto del paso 2.
4. Se aplican los operadores de agregación estadísticos seleccionados para cada uno de los conjuntos del paso 3.
5. Se identifica el descriptor generado con la siguiente nomenclatura: “fragmento químico”-“propiedad fisicoquímica” –“operador de agregación clásico”-“operador de agregación estadístico”.

2.3.3. iFeature

El software iFeature (Chen *et al.* (2018)) es una herramienta de Python que permite calcular una variedad de descriptores moleculares para proteínas y secuencias de ADN; es capaz de calcular y extraer un amplio espectro de 18 grandes esquemas de codificación de secuencias que abarcan 53 tipos diferentes de descriptores de características. También permite a los usuarios extraer propiedades específicas de los aminoácidos de la base de datos. Los grupos de descriptores son los siguientes: composición de aminoácidos, composición de aminoácidos agrupada, Pseudo composición de aminoácidos y quasi orden de secuencias. Se utilizaron únicamente los descriptores que son utilizables en secuencias de cualquier longitud. En la Tabla 3 se muestran los grupos de descriptores implementados en iFeature.

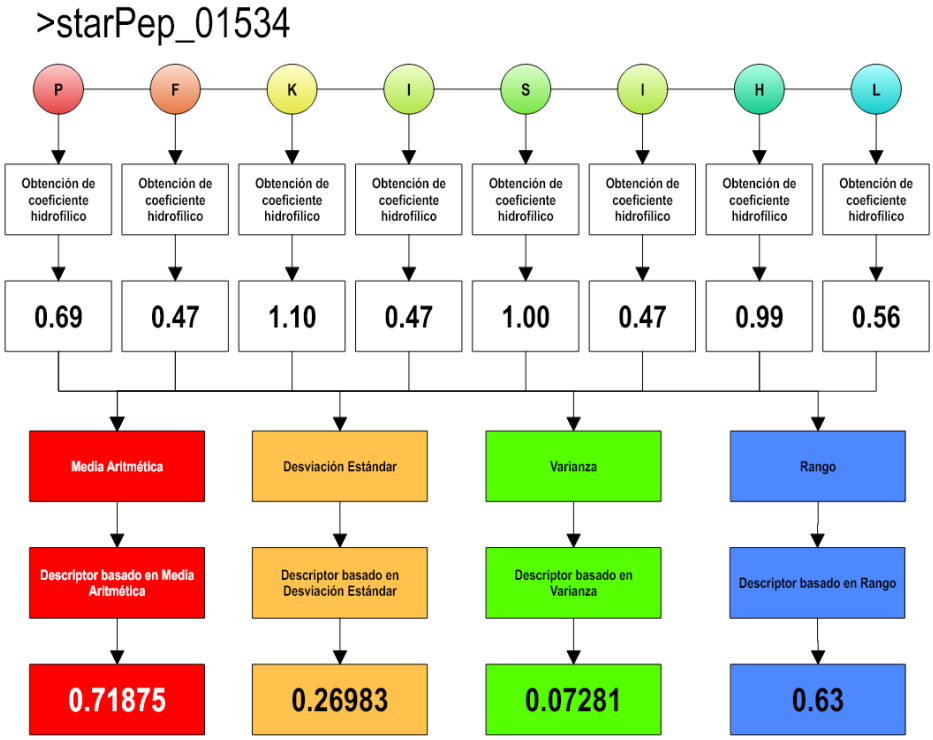


Figura 1. Ejemplo de cálculo de descriptores starPep para la secuencia PFKISIHL. Se toman los valores del coeficiente hidrofóbico de los aminoácidos de la secuencia y se utilizan diferentes operadores de agregación como la desviación estándar y la varianza.

2.3.3.1. Cálculo de descriptores de iFeature

A diferencia de software como starPep y ProtDcal, "iFeature" no posee un algoritmo único para el cálculo de descriptores moleculares, sino que este incluye una variedad de algoritmos que son ampliamente utilizados en la literatura para calcular propiedades de péptidos y proteínas. Para ejemplificar el cálculo de descriptores de iFeature se eligió el algoritmo de pseudo composición de aminoácidos, dado que es uno de los más utilizados en la literatura y muchos de los demás descriptores incluidos en este software se calculan de manera similar o están directamente basados en este. A continuación, se presentan los pasos para calcular la pseudo composición de aminoácidos:

1. Se seleccionan los parámetros λ y w que corresponden a la longitud de la correlación y al coeficiente para los valores fisicoquímicos, respectivamente. El valor de λ debe ser un valor entero positivo, que debe ser menor a la longitud de la secuencia analizada; el valor de w debe ser un valor real mayor que cero y menor o igual a uno.

Tabla 3. Descriptores de iFeature. Se enlistan los tipos de descriptores moleculares utilizados por iFeature y el número de descriptores por grupo.

Grupo de descriptores	Número de descriptores
Amino acid composition	6
Grouped amino acid composition	5
Binary	1
Autocorrelation	3
C/T/D	3
Conjoint Triad	2
Pseudo-amino acid composition	2
K-nearest neighbor	2
PSSM	1
AAindex	1
BLOSUM62	1
Z-scale	1
Predicted secondary structure	2
Predicted protein disorder	3
Predicted accessible surface area	1
Predicted main-chain torsional angles	1
Pseudo K-tuple reduced amino acids composition	16
Total	53

2. Se calcula la composición de aminoácidos de la secuencia analizada. Esto consiste en obtener la frecuencia normalizada de la ocurrencia de cada uno de los 20 aminoácidos naturales en la secuencia.

3. Se obtienen los valores normalizados de las propiedades fisicoquímicas a utilizarse. En la primera versión del algoritmo (Chou, 2009) se incluyen las propiedades de: masa de la cadena lateral, coeficiente hidrofóbico y coeficiente hidrofílico.

4. Se calcula los primeros 20 valores del vector de representación, el cual incluye la información de la composición de aminoácidos normalizada utilizando la sumatoria de los valores de propiedades fisicoquímicas (véase ecuación 3).

5. Se calculan los vectores de representación de la correlación tamaño k, que va del valor 21 a 20 + λ. Para normalizar estos valores se utiliza también la información de la composición de aminoácidos y de los valores fisicoquímicos (véase ecuación 4).

$$\frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{k=1}^{\lambda} \tau_k}, \qquad (1 \leq u \leq 20) \tag{3}$$

$$\frac{w\tau_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{k=1}^{\lambda} \tau_k}, \qquad (20+1 \leq u \leq 20+\lambda) \tag{4}$$

donde f_u es la frecuencia relativa de cada uno de los veinte aminoácidos, w es el factor de ponderación, y τ_k el factor de correlación de nivel que refleja la correlación de orden de secuencia entre los k residuos más contiguos.

2.3.4. BERT ESM

BERT ESM (Rives *et al.* (2019)) es una red neuronal desarrollada por el laboratorio de IA de Facebook, el modelo está basado en BERT (Devlin *et al.* (2019)), el cual está enfocado en el procesamiento de lenguaje natural. La aplicación de BERT ESM se entrenó con secuencias de proteínas para que esta pudiera reconocer propiedades de estas únicamente utilizando la información de la secuencia de aminoácidos.

BERT es una arquitectura basada en el modelo transformador, el cual es un mecanismo de atención que aprende las relaciones contextuales entre las palabras de un texto. En su forma simple, el transformador incluye dos mecanismos separados: un codificador que lee el texto de entrada y un decodificador que produce una predicción para la tarea. Dado que el objetivo de BERT es generar un modelo lingüístico, solamente es necesario el mecanismo codificador. En la Figura 2 se muestra el modelo transformador y BERT.

Para el entrenamiento de la red se utilizan dos fases de entrenamiento, una fase denominada pre-entrenamiento y una segunda fase denominada ajuste fino (ver Figura 3). Durante la fase de pre-entrenamiento, se utiliza una metodología de enmascaramiento de lenguaje, en el cual al modelo se le presenta una serie de frases que contienen palabras removidas (máscaras). Después el modelo intenta predecir las palabras enmascaradas, basado en el contexto del restante no enmascarado de palabras; en esta etapa del proceso el entrenamiento se realiza de manera no supervisada. En el caso particular de la red BERT-ESM, al estar utilizando únicamente secuencias proteicas para el entrenamiento, lo que se enmascaró fue un porcentaje de los aminoácidos de cada secuencia utilizada.

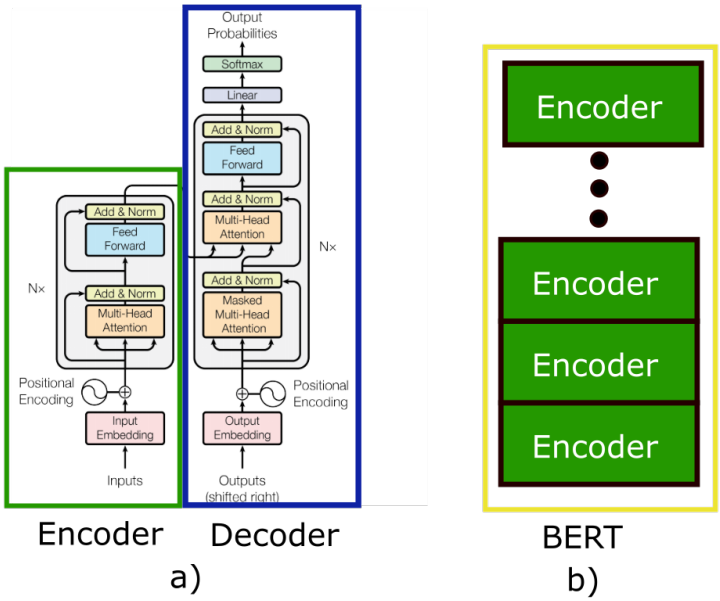


Figura 2. Diferencia entre el modelo transformador y BERT. a)Modelo Transformador (Devlin *et al.* (2019) b)Pila de codificadores utilizados en BERT.

Durante la fase ajuste fino, se entrena la red para realizar una predicción sobre la codificación del lenguaje en la etapa de pre-entrenamiento. En el caso de la aplicación de BERT al problema de procesamiento del lenguaje, un ejemplo de ajuste fino seria tomar la codificación obtenida en el pre-entrenamiento y mapearla a una traducción del mismo lenguaje o a un sistema de predicción de frases consecuentes. En el caso de BERT-ESM, se utilizó un ajuste fino para obtener un mapa de contactos de la proteína utilizada, a partir de la representación inicial.

Para utilizar BERT-ESM a manera de descriptores moleculares, únicamente es necesario utilizar la representación obtenida en el pre-entrenamiento, dado que esta representación es la que contiene la codificación de la estructura y propiedades fisicoquímicas de las proteínas, las cuales se obtuvieron a partir del aprendizaje no supervisado sobre las secuencias proteicas.

2.4. Selección de características

En el contexto del aprendizaje máquina (machine learning), se le llama selección de características al proceso de reducir el número de variables de entrada al desarrollar un modelo predictivo. Mediante este proceso se busca mejorar el desempeño del mo-

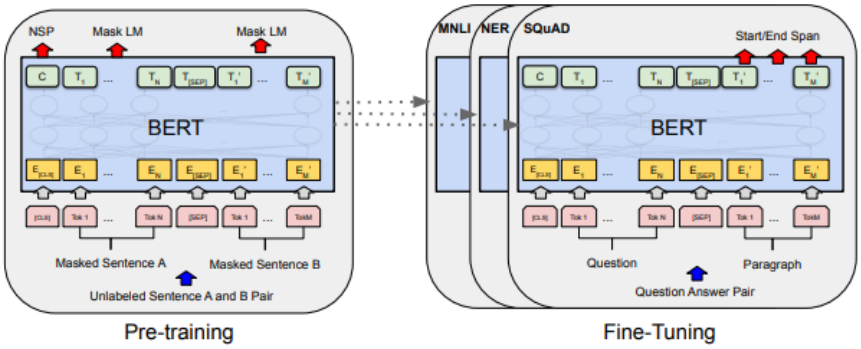


Figura 3. Proceso de pre-entrenamiento y ajuste fino, recuperado de Devlin *et al.* (2019). En la etapa de pre-entrenamiento, se entrena el modelo BERT con muestras "enmascaradas" de palabras, donde se espera que este sea capaza de predecir las palabras faltantes. En la etapa de ajuste fino, se toma esta codificación y se mapea a un lenguaje diferente.

delo que utiliza estas características, en específico se busca: mejorar la capacidad de generalización, aumentar la velocidad de aprendizaje y la reducción de la complejidad del modelo (Kumar (2014)).

Kumar (2014) platea la siguiente definición del problema de selección de características: sea A, el conjunto original de características y L, un criterio de evaluación a ser optimizado, de forma que se defina como $L : A' \subset A \rightarrow \Re$. El subconjunto de características candidato puede ser encontrado bajo las siguientes consideraciones:

- Sea $|A| = m$ y $|A'| = n$, por lo tanto, $L(A')$ es maximizada cuando $m > n$ y $A' \subset A$.
- Existe un umbral θ, tal que $L(A') > \theta$; para encontrar un subconjunto de la característica con el menor número $m > n$.
- Se puede encontrar la función de optimización $L(A')$ con subconjuntos óptimos de características A'

De la definición anterior podemos rescatar, que para que exista un proceso de selección de características debemos primero, definir un sistema de puntuación (L) que nos permita obtener un orden jerárquico de las características analizadas. A su vez, se debe definir un umbral (θ), que nos permita diferenciar las características redundantes. Al final, estas dos variables del proceso deben definirse de manera que se logre la eliminación de características, de manera que el conjunto resultante sea de una cardinalidad menor a la del conjunto inicial ($m > n$).

Existen en la literatura, una variedad de medidas que pueden ser utilizadas para medir la relevancia de las características utilizadas en la generación de un modelo. A continuación presentamos las dos metodologías de filtrado utilizadas en la metodología del Capítulo 3:

2.4.1. Filtro de entropía de Shannon

La entropía de Shannon es una medida que permite cuantificar la cantidad de información que se encuentra codificada en un mensaje independientemente de su longitud, medio o código (Shannon (1948)). En su aplicación a los descriptores moleculares, la entropía de Shannon nos permite detectar aquellos descriptores que codifican más información dentro de la distribución de moléculas que se están analizando, dado que aquellos descriptores que presentan un mayor grado de entropía son aquellos que permiten discriminar en mayor medida las diferencias estructurales de los péptidos estudiados. Se utiliza la ecuación 5 para medir la entropía de Shannon:

$$H = -\sum_{i=1}^{n} P(x_i)\, log_b\, P(x_i) \tag{5}$$

donde "x_i" es una instancia que pertenece a la distribución del descriptor analizado, La función "P" es la probabilidad de la instancia y la constante "b" es la base del logaritmo que puede ser escogida arbitrariamente. Con base en esta métrica, los descriptores desechados son aquellos que presentaron la menor entropía.

2.4.2. Filtro correlación de Spearman

Dos variables pueden estar relacionadas en una relación no lineal, que puede ser más fuerte o más débil en la distribución de las variables. Una medida de esta relación es el coeficiente de correlación de Spearman, esta es una prueba no paramétrica que se usa para medir el grado de asociación entre dos variables con una función monótona, es decir, una relación creciente o decreciente (Zar, 2005). En su aplicación a los descriptores moleculares, esta medida nos dice cuáles descriptores tienen una relación lineal en sus distribuciones. En este caso, nos interesa que nuestros descriptores

no sean linealmente similares entre sí, debido a que esto puede considerarse como información redundante para el proceso de clasificación. Se utiliza la ecuación 6 para medir la correlación de Spearman:

$$\rho = 1 - \frac{6\sum(x_i - y_i)^2}{n(n^2-1)} \tag{6}$$

donde "x_i" e "y_i" representan las distribuciones analizadas; y "n", representa el número de muestras.

2.5. Modelado

Como se definió en la sección 2.1, un modelo QSAR es aquel que correlaciona las propiedades fisicoquímicas y estructurales de las moléculas con una determinada actividad. Un modelo puede definirse como se hizo en la ecuación 2, donde la actividad está en función de las propiedades. Durante el proceso de modelado, lo que se busca es encontrar dicha función, que tome como entrada los descriptores moleculares seleccionados y mapee estos a una decisión, comúnmente si la molécula analizada presenta o no determinada actividad.

En el contexto de aprendizaje de máquina, se le conoce como clasificación al proceso anteriormente descrito. Podemos definir formalmente el proceso de clasificación como: el proceso de tomar como entrada un vector "x", y asignarlo a una clase discreta "C_k". El espacio de entrada que se genera por las instancias utilizadas en el modelo, se divide en regiones llamadas "regiones de decisión" cuyos límites se denominan "fronteras de decisión" o "superficies de decisión" (Bishop, 2006). El modelado predictivo de la clasificación es la tarea de aproximar una función de mapeo "f" del vector de entrada "x", a las variables de salida discretas "C_k". Matemáticamente se define como:

$$C_k = f(x_1, x_2, x_3, ..., x_n) \text{ donde } C_k \in \{0, 1, 2, ..., k\} \tag{7}$$

en el contexto de la clasificación de péptidos antimicrobianos, los valores de x_n son

los descriptores moleculares; $f(x_n)$ es el modelo de aprendizaje máquina que modela la actividad biológica; y C es el conjunto de clases a las cual se puede asignar una instancia. Existen varios paradigmas de clasificación, siendo el más común para la clasificación de péptidos antimicrobianos la clasificación de dos clases; sin embargo también se ha utilizado la clasificación multietiqueta, y en este documento introducimos la clasificación de una clase, aplicada a este problema. A continuación se presenta la definición de cada una de los paradigmas de clasificación, aplicada a la clasificación de péptidos antimicrobianos.

2.5.1. Clasificación binaria de AMP

La clasificación binaria de AMP se define como el problema de decidir si una secuencia peptídica, con base en su descripción molecular, pertenece o no al conjunto AMP. Se define como la aplicación de la ecuación 8, cuando las clases se definen como sigue:

$$C \in \{AMP, NO-AMP\} \tag{8}$$

donde C es la clase a la que puede pertenecer una instancia, en el caso de la clasificación binaria, una instancia puede pertenecer solo a una de las dos clases. Para resolver este problema de clasificación se han utilizado algoritmos como máquina de soporte vectorial (Joseph *et al.* (2012) y Ng *et al.* (2015)), bosque aleatorio (Thomas *et al.* (2010) y Bhadra *et al.* (2018)), K vecinos más cercanos (Xiao *et al.* (2013)), redes neuronales profundas (Veltri *et al.* (2018)).

2.5.2. Clasificación multietiqueta de AMP

Los péptidos antimicrobianos pueden tener varias funciones (antibacteriales, antivirales, antifúngicas, anticancerígenas, insecticidas, etc.) e incluso pueden existir péptidos que tengan más de un tipo de funcionalidad. Por lo tanto se define la clasificación

multietiqueta de péptidos como sigue (ver ecuación 9):

$$C \subseteq \{Antibacterial, Antiviral, Antifngico, Anticncer, Antiparasitario, ..., n\}, \quad |C| \leq n \tag{9}$$

donde C es la clase a la que puede pertenecer una instancia, en el caso de la clasificación multietiqueta, una instancia puede pertenecer a una o más clases ($|C| \leq n$). La clasificación multietiqueta de péptidos antimicrobianos se puede definir como una extensión de la clasificación binaria a más de dos clases, por lo tanto, para la clasificación se utilizan algoritmos similares a los utilizados en la clasificación binaria. Para resolver este problema, se utilizan los algoritmos mencionados en la sección 2.5.1, adaptados para la clasificación multiclase.

2.6. Clasificación de una sola clase

En la clasificación de una clase, el problema consiste en clasificar los datos cuando se dispone de información para un solo grupo de observaciones. Concretamente, dado un conjunto de datos, denominado clase objetivo, el objetivo de los métodos de la clasificación de una clase es distinguir los datos pertenecientes a la clase objetivo de otras clases posibles. La clasificación de una clase puede considerarse un tipo especial de problema de clasificación de dos clases, cuando se consideran los datos de una sola clase. Se trata de un problema interesante porque hay muchas situaciones reales en las que obtener un conjunto representativo de ejemplos etiquetados para la segunda clase tiene un costo elevado, es difícil de obtener o no está disponible en absoluto (Irigoien *et al.* (2014)).

Dada la naturaleza del enfoque, la clasificación de una clase es más adecuada para aquellas tareas en las que los casos positivos no tienen un patrón o estructura consistente en el espacio de características, lo que dificulta que otros algoritmos de clasificación determinen un límite para la clase objetivo. En cambio, tratar los casos negativos como valores atípicos, permite a los clasificadores de una clase ignorar la tarea de discriminación y centrarse, en cambio, en las desviaciones de lo normal a lo esperado.

Según Perera *et al.* (2021), la clasificación de una clase se puede definir en tres vertientes: la detección de novedad (novelty detection), la detección de anomalías (outlier detection) y el reconocimiento de conjunto abierto (open set recognition). A continuación se describen las tres vertientes.

2.6.1. Detección de novedad

En este caso, el clasificador define unos límites en el espacio de características para las instancias dadas; si existe una muestra cuya representación exista dentro de estos límites, se considera que dicha muestra pertenece a la clase objetiva, de otra manera, se considera la muestra como una anomalía (outlier). En la Figura 4, se muestra una gráfica de un método de detección de novedad.

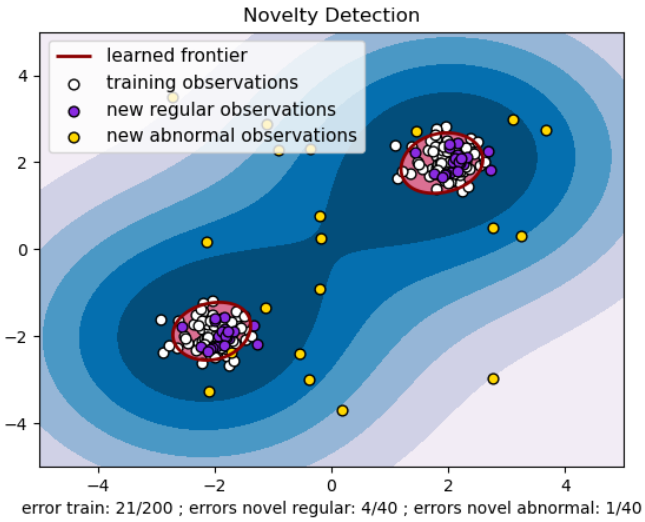


Figura 4. Ejemplo de un algoritmo de detección de novedad. Se muestra el espacio de conocimiento generado por un clasificador de una sola clase, así como la predicción sobre nuevas observaciones. Imagen recuperada de la página oficial de Sklearn: https://scikit-learn.org/.

Como podemos observar en la Figura 4, las nuevas observaciones regulares, son aquellas que caen dentro del espacio de conocimiento definido por el modelo; todas las demás observaciones, se consideran anormales o anómalas.

2.6.2. Detección de anomalías

En detección de anomalías, se presenta una mezcla de datos normales y datos anormales, sin conocer o tener alguna referencia de qué dato podría pertenecer a cuál clase. El objetivo es separar los datos normales de los anormales utilizando técnicas no supervisadas. En cambio, en la detección de novedad se presupone que todos los datos de entrenamiento son normales. Por lo tanto, la detección de novedad se considera un problema de aprendizaje supervisado, mientras que la detección de anomalías es un problema no supervisado. En la Figura 5, se muestra como ejemplo el algoritmo Local Outlier Factor (Breuniq *et al.* (2000)), en este ejemplo los radios de los círculos rojos están en función de las puntuaciones del algoritmo, de manera que entre más grande el círculo mayar la probabilidad tiene la instancia de ser “outlier”.

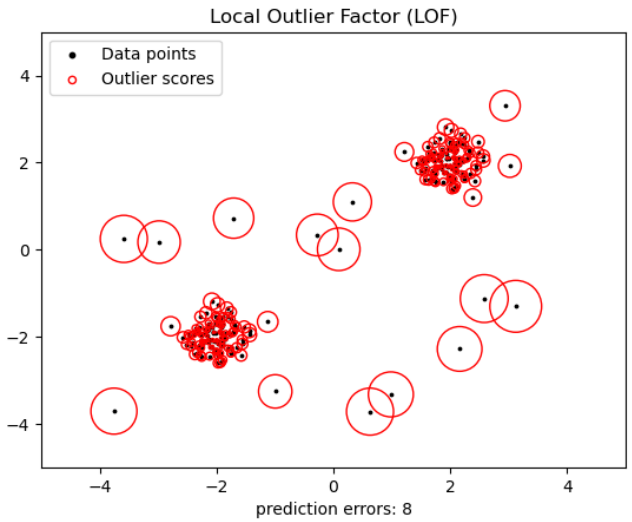


Figura 5. Ejemplo del algoritmo de detección de anomalías: Local Outlier Factor. Se muestra el grado de “outlier” de cada instancia como un círculo rojo, entre más “outlier”, más grande el círculo. imagen recuperada de la página oficial de Sklearn: https://scikit-learn.org/.

El algoritmo “Local Outlier Factor”, utiliza únicamente las relaciones de distancias entre una instancia del conjunto dado en relación con las demás instancias para inferir si una determinada muestra se encuentra aislada o no. Existen muchos ejemplos de algoritmos de detección de anomalías que pueden ser transformados en métodos supervisados, como es el caso del mismo algoritmo de “Local Outlier Factor”; en estos casos se utiliza el sistema de decisión del algoritmo de detección de anomalías para probar instancias nuevas, a manera de un algoritmo de detección de novedad.

2.6.3. Reconocimiento de conjunto abierto

El reconocimiento de conjuntos abiertos es una extensión de la clasificación de una clase, a la clasificación multiclase. En este caso, se plantea la construcción de un modelo de clasificación de una clase, para cada una de las etiquetas que se utilizan en la clasificación multiclase. En la Figura 6, se muestra un ejemplo de reconocimiento de conjunto abierto.

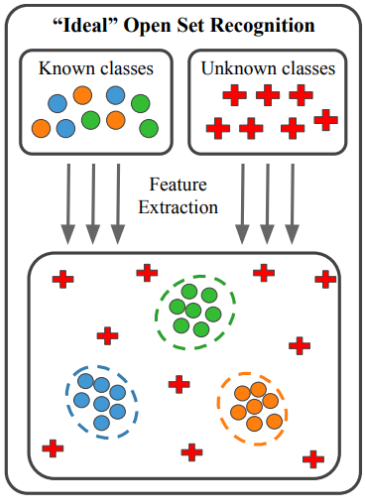


Figura 6. Ejemplo del algoritmo de reconocimiento de conjunto abierto. En esta metodología se entrena un clasificador de una clase, por clase. Esto permite realizar clasificación multiclase utilizando clasificadores de una clase. Imagen recuperada de Miller *et al.* (2021).

En la Figura 6, se muestra cómo para cada uno de las posibles clases del conjunto de datos se generó un modelo de clasificación de una clase. Por lo tanto, cada instancia debe probarse en cada uno de los modelos obtenidos, de forma que si no es recuperada por alguna de estas, la muestra se considera anómala.

2.6.4. Ejemplos de clasificadores de una sola clase

2.6.4.1. Isolation Forest

El algoritmo de "Isolation Forest", al igual que otros métodos de ensamble de árboles, se construye con base en árboles de decisión (Liu *et al.* (2008)). En estos árboles primero se crean particiones seleccionando aleatoriamente una característica y luego seleccionando un valor de partición aleatorio entre el valor mínimo y máximo de la característica seleccionada (ver Figura 7).

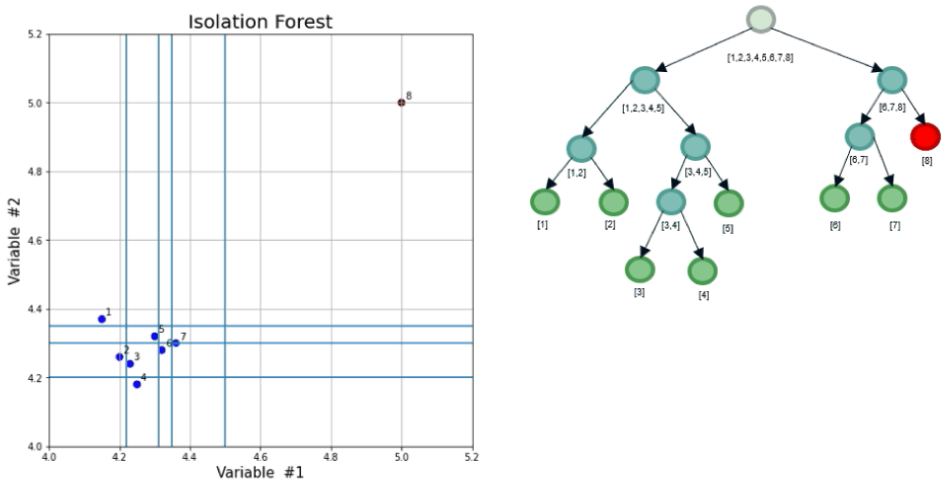


Figura 7. Ejemplo de las particiones binarias generadas por el algoritmo Isolation Forest. Las particiones binarias se muestra sobre el conjunto de puntos, mientras que el orden de las particiones se muestra en el árbol binario. Realizado con Matplotlib y Microsoft PowerPoint.

En el ejemplo de la Figura 7, se muestra una serie de particiones aleatorias sobre un conjunto de datos generado de manera aleatoria. En la gráfica de la izquierda en la Figura 7, las particiones están representados por las barras azules que interceptan los planos *X* (Variable # 1) e *Y* (Variable #2), mientras que en el árbol de la derecha, están representadas por los nodos. En el árbol de la Figura 7 se puede observar la serie de particiones que siguió el algoritmo de “Isolation Forest” para aislar a todos los nodos, al utilizarse un método de particiones aleatorias, el algoritmo puede generar una serie de árboles con diferentes longitudes.

En principio, los valores atípicos (anomalías) son menos frecuentes que las observaciones regulares y se diferencian de ellas debido a que están más alejados de las observaciones regulares en el espacio de características. Por ello, al utilizar particiones aleatorias, es más probable que las observaciones de valores atípicos se encontraran más cerca de la raíz del árbol debido a que necesitan menos particiones para aislarse. Dado que las anomalías son susceptibles al aislamiento y tienen la tendencia de residir más cerca de la raíz del árbol de decisión (ver Figura 8), se construye el árbol de decisión hasta que alcanza una cierta altura máxima, considerando que a partir de que se alcanza esta altura es poco probable encontrar más muestras anómalas (Liu *et al.* (2008)).

Al igual que con otros métodos de detección de anomalías, se requiere de un sis-

tema de puntuación para la toma de decisiones. En el caso del Isolation Forest, el sistema de puntuación se define como sigue:

$$S(x, n) = 2^{-E(H(x))/c(n)} \tag{10}$$

donde E(H(x)) es la longitud promedio de la trayectoria media de H(x) en un conjunto de árboles de aislamiento, c(n) es el promedio de la longitud del trayecto H(x) dado *n* y *n* es el número de nodos externos (Liu *et al.* (2008)). De la función de puntuación definida anteriormente, podríamos deducir que:

- si el puntaje de una muestra es muy cercano a 1, entonces definitivamente es una anomalía.
- si la puntuación de las muestras analizadas es mucho menor que 0,5, entonces es muy probable que estas sean observaciones regulares.
- Si todas las puntuaciones se acercan a 0,5, entonces la muestra analizada no tiene anomalías detectables.

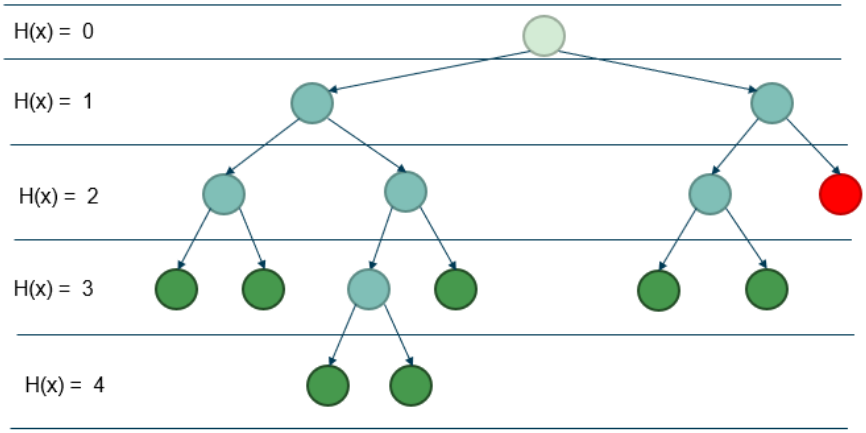


Figura 8. Ejemplo de la función H(x) generada por el algoritmo Isolation Forest. El algoritmo puntúa a las instancias con base en qué tan fácil fueron de aislar, si estas se aislaron con facilidad, aparecerán más arriba en el árbol binario.

2.6.4.2. Local Outlier Factor

Local Outlier Factor (LOF), es un algoritmo no supervisado utilizado para la detección de anomalías (outliers). Produce una puntuación midiendo la desviación de la

densidad local de un punto en el conjunto de datos con respecto a los puntos de datos cercanos a él (Breuniq *et al.* (2000)).

La densidad local se determina estimando las distancias entre los puntos que son vecinos (k-vecinos más cercanos). Así que para cada punto la densidad local puede ser calculada. Al compararlas podemos comprobar qué puntos en el conjunto de datos tienen densidades similares y cuáles tienen una densidad menor que sus vecinos. Los que tienen las menores densidades son considerados como los atípicos (Breuniq *et al.* (2000)).

Primero, se empieza por calcular la “distancia de alcance” (reachability distance) para cada combinación de puntos en el conjunto de datos. La distancia de alcance puede ser definida como:

$$RD(X_i, X_j) = max(K - distancia(X_j), distancia(X_i, X_j)) \quad (11)$$

donde los valores de X son dos puntos cualquiera del conjunto de datos, distancia es una medida de distancia cualquiera (Euclidiana, Manhattan, etc.) y la K-distancia es la distancia del punto X_j a su “K” vecino más cercano (ver Figura 9); la ecuación toma el valor del valor máximo entre estas dos medidas (Breuniq *et al.* (2000)).

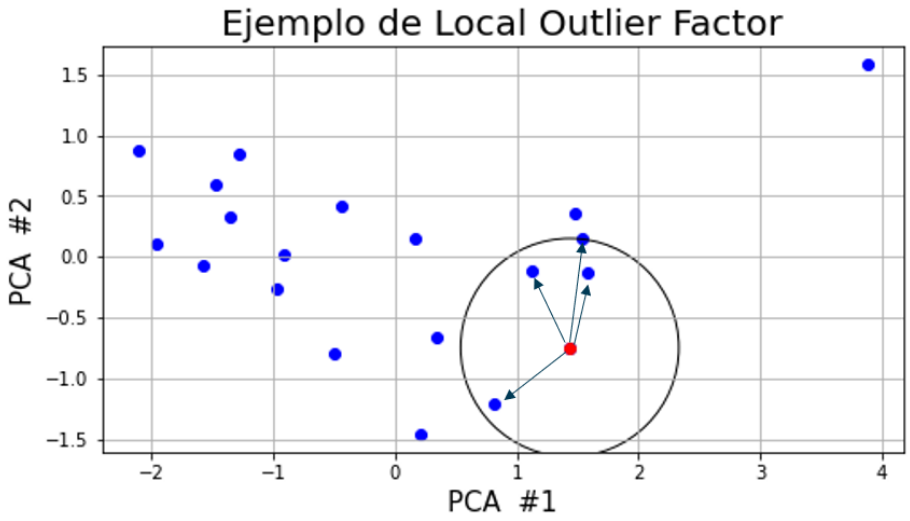


Figura 9. Ejemplo de la distancia de alcance para K=4, en el conjunto de puntos semialeatorio. En esta figura, el círculo negro representa la distancia local de alcance para la instancia resaltada en color rojo.

Una vez calculada la distancia de alcance de todos los puntos del conjunto de datos procedemos a calcular la densidad de alcance local (Local Reachability Density). La densidad de alcance local es una medida de la densidad de los puntos más cercanos a un punto que se calcula tomando el inverso de la suma de todas las densidades de todos los puntos vecinos más cercanos. Cuanto más cercanos están los puntos, la distancia es menor, y la densidad es mayor, por lo que se toma la inversa en la ecuación 9. La densidad de alcance local está definida por la siguiente ecuación:

$$LRD_k(A) = 1/\sum_{X_j \in N_k(A)} RD(A, X_j)/N_k(A) \tag{12}$$

la densidad de alcance local de cada punto se usa para comparar con la densidad de alcance local promedio de sus k-vecinos.

El Local Outlier Factor es la proporción de la densidad promedio de los k-vecinos de A con la densidad de A, donde A es una muestra del conjunto de datos. El Local Outlier Factor está definido por la siguiente ecuación:

$$LOF_k(A) = \sum_{X_j \in N_k(A)} (LDR_k(X_j)/N_k(A)) * (1/LDR_k(A)) \tag{13}$$

si la densidad de los vecinos y el punto son casi iguales podemos decir que son bastante similares; si la densidad de los vecinos es menor que la densidad del punto podemos decir que el punto es un atípico, es decir, dentro del cúmulo, y si la densidad de los vecinos es mayor que la densidad del punto podemos decir que el punto es un atípico. El sistema de puntuación de LOF se puede interpretar como sigue:

- si LOF $\approx$ 1 es un punto que probablemente sea una observación regular.
- si LOF $<$ 1 el punto es una observación regular.
- si LOF $>$ 1 el punto se considera anómalo.

En la Figura 10 se muestra el cálculo del Local Outlier Factor para el conjunto aleatorio de la Figura 9. Aquí podemos observar que la muestra más alejada, observada en

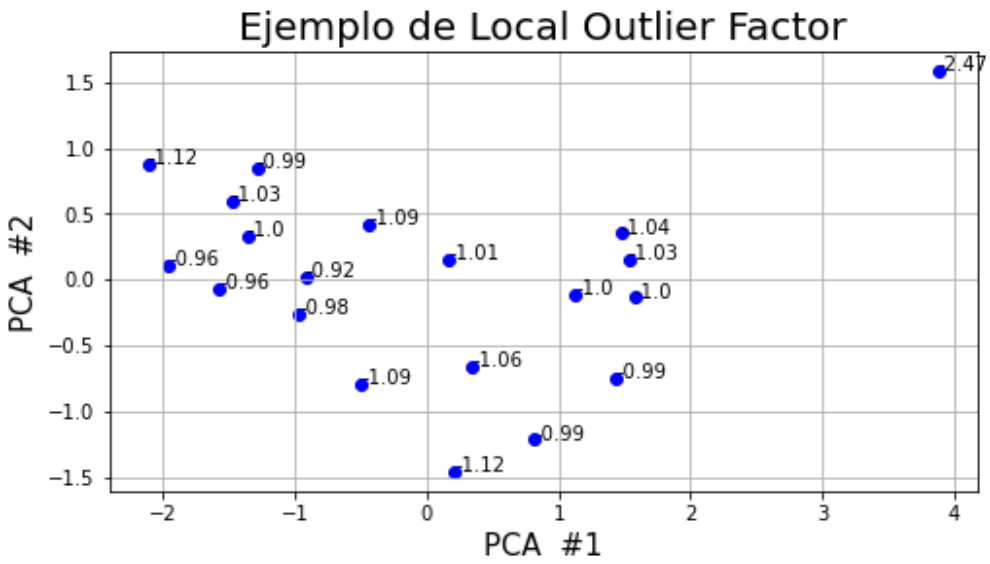


Figura 10. Ejemplo del cálculo del Local Outlier Factor para K=4, en un conjunto de puntos semialeatorio de la Figura 9. Los números que se muestran en cada instancia, son las puntuaciones calculadas por el algoritmo de Local Outlier Factor.

la esquina superior derecha, tiene una calificación mucho más alta que 1, a diferencia de las demás muestras.

2.7. Validación

Una vez que contamos con un modelo, es de interés averiguar qué tan precisa es la capacidad de predicción de este. A este proceso se lo conoce como validación de modelo, durante este se utiliza la función del modelo previamente entrenada para predecir un conjunto de moléculas que si bien pertenecen al mismo conjunto de moléculas que se está estudiando, las mismas no fueron utilizadas durante el proceso de entrenamiento. Por lo regular la efectividad del modelo se mide en función de unas cuántas instancias del conjunto de prueba se clasificaron de manera correcta. Algunas de las medidas más usadas en la literatura son: Exactitud, especificidad, sensibilidad, coeficiente de correlación de Matthews, entre otros. A continuación se presenta una definición de cada uno de estos conceptos.

- **Sensibilidad:** este valor representa la capacidad del modelo de recuperar mues-

tras pertenecientes al espacio de conocimiento del modelo (muestras positivas).

$$S = \frac{VP}{VP + FN} \tag{14}$$

- **Especificidad:** este valor representa la capacidad del modelo de filtrar muestras que no son pertenecientes al espacio de conocimiento del modelo (muestras negativas).

$$E = \frac{VN}{VN + FP} \tag{15}$$

- **Exactitud:** representa la capacidad del modelo de hacer una clasificación correcta, puede verse como el promedio entre los valores de especificidad y sensibilidad.

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \tag{16}$$

- **Coeficiente de correlación de Matthews:** el coeficiente tiene en cuenta los verdaderos y falsos positivos y negativos, y suele considerarse una medida equilibrada que puede utilizarse incluso si las clases son de tamaños muy diferentes.

$$MCC = \frac{(VP * VN) - (FP * FN)}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \tag{17}$$

donde VP son los verdaderos positivos, VN son los verdaderos negativos, FP son los falsos positivos y FN los falsos negativos.

2.8. Dominio de aplicabilidad

Según Roy *et al.* (2015), se define el dominio de aplicabilidad como una región teórica en el espacio químico construido tanto por los descriptores del modelo como por la respuesta del modelo. El dominio de aplicabilidad puede ser definido también como la región del espacio químico para la cual nuestro modelo tiene validez (Todeschini y Consonni, 2009), más allá de esta región no se tiene certeza de la validez del modelo. Por lo tanto, la predicción de una actividad modelada usando QSAR solamente es aplicable si el compuesto que se predice se encuentra dentro del dominio, ya que es inviable predecir todo el universo de compuestos utilizando un solo modelo QSAR (Roy *et al.* (2015).

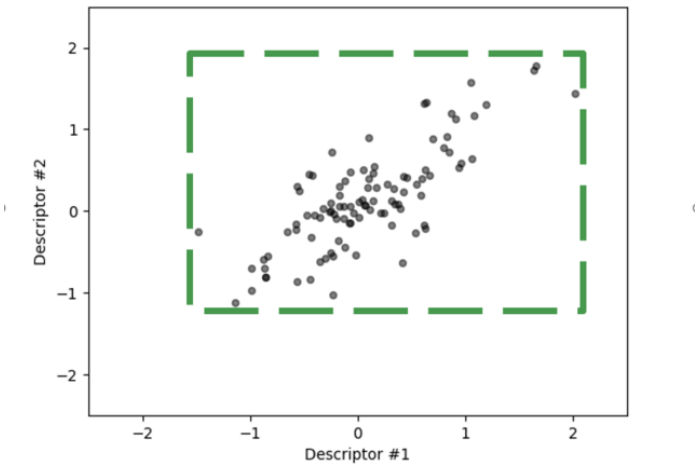


Figura 11. Dominio de aplicabilidad generado utilizando el intervalo de los descriptores moleculares. En este caso, el dominio de aplicabilidad está definido por el máximo y el mínimo del descriptor #1 y el descriptor #2.

La primera limitante para la aplicabilidad de un modelo a una determinada molécula, es que esta pertenezca a la misma clase de moléculas con la que fue entrenado el modelo (Hanser *et al.* (2019)). Un ejemplo puede ser un modelo capaz de aceptar moléculas orgánicas, pero en su diseño está el excluir polímeros, proteínas y moléculas inorgánicas. Otra limitante para los modelos QSAR puede ser el conjunto de descriptores moleculares que definen al conjunto de entrenamiento, en este caso, podría ser imposible calcular los descriptores moleculares para una molécula “X” debido a que su estructura no posee alguna característica que si poseen las moléculas de entrenamiento.

El dominio de aplicabilidad está definido por la representación de las moléculas dado por el conjunto de descriptores moleculares seleccionados. Una metodología común para la definición de un espacio de aplicabilidad es la de tomar el intervalo completo de cada uno de los descriptores moleculares y definir el dominio como este espacio. Un ejemplo puede verse en la Figura 11, donde inicialmente se tiene un conjunto de moléculas representadas por dos descriptores moleculares, en este caso, el dominio de aplicabilidad es el área delimitada por el rectángulo verde, definida por el intervalo de los descriptores tanto en el eje “x” como en el eje “y”.

En la aplicación de modelados estadísticos o de aprendizaje máquina para la modelación QSAR, no es común que un modelo recupere todo el intervalo de valores

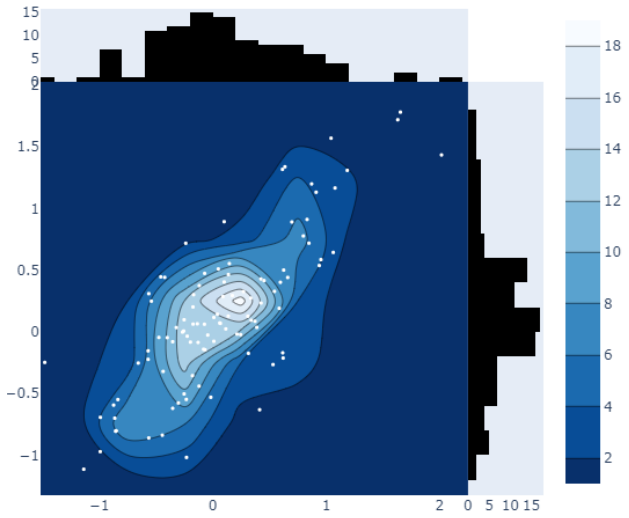


Figura 12. Dominio de aplicabilidad generado por la densidad de datos. En esta imagen se define el dominio de aplicabilidad en función de la densidad de datos, estando este limitado al espacio con mayor densidad de datos.

de descriptores moleculares, sino más bien, se recuperan únicamente las instancias donde existe una determinada densidad de observaciones.

En el ejemplo de la Figura 12, se puede observar un mapa de densidad sobre el mismo conjunto de la Figura 11. En este caso, un modelo QSAR construido con estos datos, únicamente podría recuperar muestras con un determinado grado de confianza, en las regiones del espacio con una densidad de 10 o mayor, dado que estas son las regiones con más densidad de datos. Entre más se alejan los datos de la región central, podemos observar que estas poseen una menor cantidad de muestras, por lo tanto, es una región en la cual se tiene una menor confianza estadística (Hanser *et al.* (2019)).

Con base en lo anterior, podemos concluir que el dominio de aplicabilidad de un modelo QSAR, es una región del espacio de descriptores donde se tiene una alta confianza estadística en la recuperación del modelo, y que está soportado por la densidad de muestras que contiene. Entre más crítica sea la aplicación del modelo QSAR, mayor debe ser la evidencia que soporta la decisión del modelo.

2.9. Péptidos

Un péptido puede ser definido como un polímero generado por la unión secuencial de aminoácidos. Los aminoácidos son moléculas que están conformadas por un átomo de carbono que se denomina “carbono alfa” y que es el centro de la molécula. En los enlaces del carbono alfa se encuentran siempre: un grupo amino ($-NH_2$), un grupo carboxilo ($-COOH$) y un grupo funcional ”R”. Los aminoácidos pueden ser diferenciados por su grupo funcional, siendo 20 los que comúnmente intervienen en los procesos biológicos, siendo estos denominados como aminoácidos naturales o canónicos (Nelson y Cox, 2008).

Para la formación de un péptido, dos o más aminoácidos se unen por medio de un enlace peptídico. En el enlace peptídico, el extremo amino ($-NH_2$) de un aminoácido se une con el extremo carboxilo ($-COOH$) de otro aminoácido, generando un enlace covalente y agua como subproducto de la reacción. A la a unión de entre 2 y 20 aminoácidos se le como “oligopéptido” y si se supera esta longitud, se le conoce como “polipéptido” (Alberts *et al.* (2002)). Los péptidos miden entre 2 y 100 aminoácidos de longitud, si se supera esta medida, se considera una proteína.

Con el crecimiento de las cadenas de péptidos, estas tiende a plegarse en diferentes formas, siendo las más comunes la “alfa hélice” y las “láminas beta”. En el caso de la alfa hélice, esta estructura es generada por una unión mediante puentes de hidrógeno, en particular, el grupo carboxilo de cada aminoácido forma un enlace de hidrógeno con el grupo amino del aminoácido que está situado cuatro residuos por delante en la secuencia. Las láminas beta, también se forman por medio de puentes de hidrógeno, pero en este caso, dos o más segmentos de una cadena polipeptídica se alinean uno junto a otro, formando una estructura laminar (Berg *et al.* (2007)).

Gracias a la variedad de estructuras y de propiedades fisicoquímicas que presentan los péptidos, es que estos pueden adquirir una serie de funciones biológicas diferentes. Entre estas funciones podemos encontrar: funciones hormonales, neurotransmisores, comunicación, catálisis, e inhibición de organismos patógenos (Nelson y Cox, 2008).

2.9.1. Péptidos antimicrobianos

Podemos definir un péptido antimicrobiano, como un péptido al cual se le atribuye la capacidad de inhibición de uno o más agente patógeno, entre los que pueden estar: bacterias, virus, hongos, parásitos, etc (Phoenix *et al.* (2013)). Además, se ha probado que estos tienen efectividad contra organismos que presentan resistencia a los antibióticos convencionales, lo cual los hace candidatos para combatir a los organismos resistentes (WHO, 2014). También tienden a tener baja o nula interacción con células animales debido a que son moléculas que han evolucionado para atacar patógenos específicos, lo cual los hace candidatos para desarrollar terapias basadas en ellos (Peters *et al.* (2010)).

La capacidad de los péptidos antimicrobianos (AMP) para eliminar agentes patógenos, depende principalmente de su capacidad para interactuar con la membrana citoplasmática de estos organismos (Phoenix *et al.* (2013)). En la mayoría de los casos, los microorganismos tienen una membrana compuesta por peptidoglicano, la cual está cargada negativamente (aniónica), por lo tanto, tendrán una mayor interacción con estas membranas los péptidos que poseían una carga positiva (catiónicos) (Zasloff, 2002). Según Phoenix *et al.* (2013), a raíz de estas interacciones surgen los mecanismos de acción que se describen a continuación:

- **Mecanismo de barril:** en este caso, el péptido se introduce en la membrana con una orientación perpendicular a la superficie de la bicapa, lo cual conduce a la creación de un poro transmembrana.
- **Poro toroidal:** este mecanismo de acción es similar el mecanismo de barril. Sin embargo, el modelo de poro toroidal propone que la agregación de péptidos en la superficie de la membrana impone una tensión de curvatura positiva al aumentar la distancia entre los grupos de lípidos de la membrana.
- **Mecanismo de alfombra:** en este mecanismo, los péptidos se acumulan de manera paralela a la membrana del patógeno y una vez alcanzada una concentración crítica, se alinean de manera que produzcan un poro toroidal.
- **Mecanismo de péptido inclinado:** en este mecanismo, los péptidos penetran la membrana con un ángulo de entre 20^{o} y 80^{o} respecto a la normal de la bicapa,

lo que crea una curvatura negativa en la membrana, promoviendo la alteración de los lípidos que conforman la membrana.

En esta investigación nos enfocamos en los péptidos con capacidad de inhibición de bacterias Gram positivas y Gram negativas. Las bacterias se denominan Gram positivas o Gram negativas dependiendo del color que toman después del proceso de tinción de Gram, siendo de color violeta las bacterias Gram positivas y color rosado las bacterias Gram negativas (Willey *et al.* (2020)). Las membranas internas o citoplasmáticas de ambos grupos de bacterias son similares, sin embargo, las envolturas celulares externas son significativamente diferentes. En el caso de las bacterias Gram positivas tenemos una capa de peptidoglicano con ácidos teicoicos, mientras que en el caso de las bacterias Gram negativas tenemos una capa más pequeña de peptidoglicano con una membrana externa compuesta de lipopolisacáridos (Tortora *et al.* (2019)). En este caso, la capacidad de inhibición también proviene de la capacidad de los péptidos de interactuar con las membranas de las bacterias Gram positivas y Gram negativas.

Según Li *et al.* (2017), en el caso de las bacterias Gram positivas, los péptidos necesitan difundirse a través de la matriz de peptidoglicano primero y luego actuar sobre la membrana citoplasmática. Por el contrario, la eliminación de las bacterias Gram negativas implica la perturbación o disrupción tanto de las membranas externas como de las citoplasmáticas. La incapacidad de permeabilizar o alterar la membrana externa provoca la pérdida de la actividad antimicrobiana.

Capítulo 3. Metodología

En este capítulo se presenta un procedimiento para la identificación de péptidos antimicrobiano enfocado en detectar aquellos que posean propiedades tanto Gram positivas como Gram negativas. Se utiliza una metodología jerárquica para lograr este proceso de clasificación; en el cual primero, se clasifican las muestras utilizando un modelo entrenado para diferenciar entre secuencias que poseen una actividad antimicrobiana y aquellas secuencias que no poseen esta propiedad. Después, se utiliza un modelo entrenado para clasificar entre péptidos que poseen propiedades antibacterianas y aquellos que no la tienen; por último, se utiliza un modelo entrenado para detectar muestras ya sea Gram positivo o Gram negativo. Cabe destacar que únicamente se consideran como Gram positivas o Gram negativas aquellas secuencias que hayan sido clasificadas como positivas por cada uno de los tres filtros. Un bosquejo del procedimiento propuesto se muestra en la Figura 13.

3.1. Metodología de construcción de conjuntos de entrenamiento y validación

Las secuencias que sean elegidas durante el proceso de generación de conjuntos de entrenamiento y validación, son las que definirán el espacio químico y el dominio de aplicabilidad de los modelos que las utilicen. Por lo tanto, es muy importante elegir secuencias que nos ayuden a modelar de manera correcta la actividad biológica que se está estudiando. Aunque no existe una única metodología para la obtención y procesamiento de secuencias para la generación de conjuntos, en la literatura se observa que propuestas que pretenden clasificar péptidos antimicrobianos utilizan la siguiente serie de pasos:

1. Se recuperan las secuencias con actividad antimicrobial de las bases de datos públicas como pueden ser: APD (Wang *et al.* (2016)) o DRAMP (Kang *et al.* (2019)) por mencionar algunas.
2. Utilizando herramientas como BLAST (Altschul *et al.* (1990)) o CD-Hit (Li y Godzik, 2006), se agrupan las secuencias que, entre sí, posean un porcentaje de identidad igual o mayor a un umbral arbitrario (definido por el usuario). Una vez agrupadas,

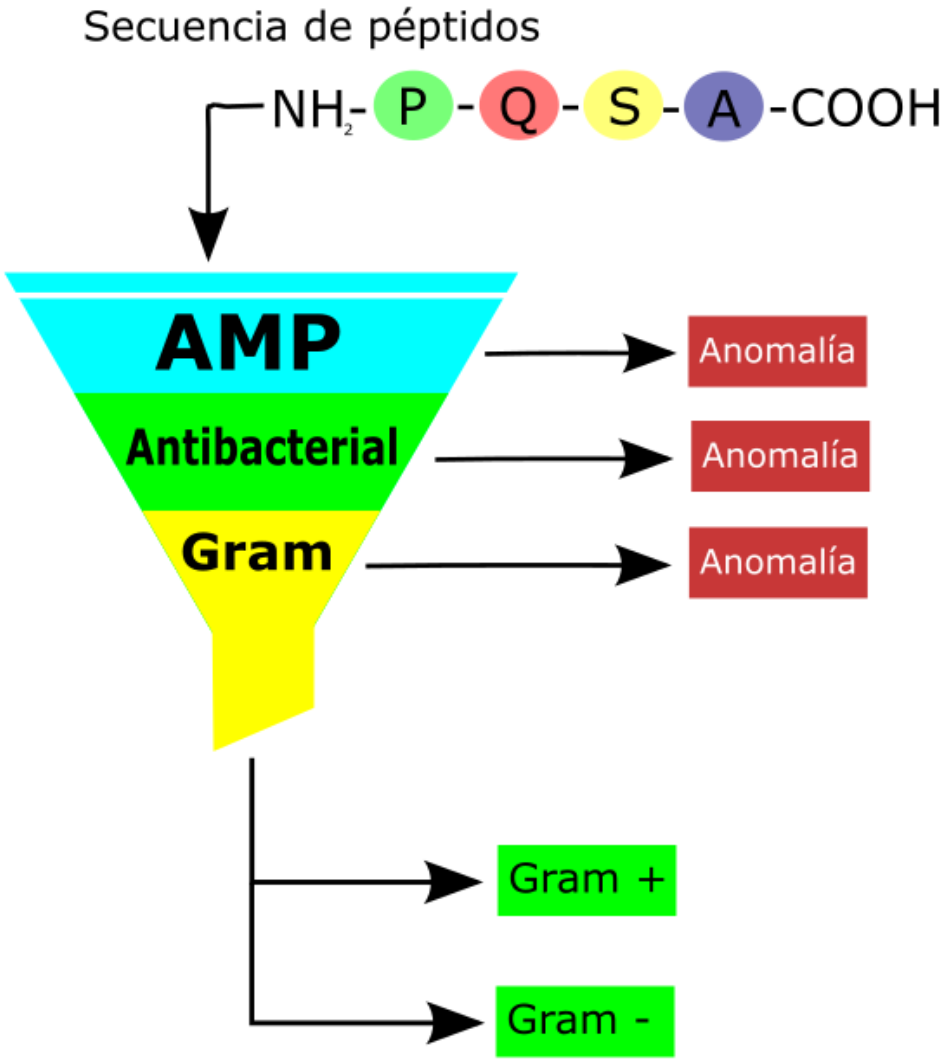


Figura 13. Bosquejo del modelo de clasificación utilizado. Se representa el modelo jerárquico donde se utilizan de manera secuencial los filtros de AMP, antibacterial, Gram positivo y Gram negativo.

se elige una secuencia de cada grupo como representante y se descartan las demás secuencias.

3. Se filtran las secuencias con base en su longitud.
4. Las secuencias se dividen en conjuntos en función de su actividad biológica.
5. Se crean conjuntos de entrenamiento y prueba dividiendo los conjuntos obtenidos en el paso número 3 utilizando un determinado criterio, en muchos ejemplos de la literatura se realiza este paso de manera aleatoria en una proporción 80/20 entre el conjunto de entrenamiento y el de prueba, respectivamente.

Los pasos mencionados anteriormente son una generalización de lo que puede ser encontrado en la literatura, algunos ejemplos de la literatura que mencionan esta metodología o una similar son los trabajos de: Xiao *et al.* (2013), Veltri *et al.* (2018), Torrent *et al.* (2009), y Pinacho-Castellanos *et al.* (2021a), entre otros. Los pasos para generar los conjuntos de validación negativos (No-AMP) se mencionan en la sección 3.1.2.

Para desarrollar la metodología QSAR para la clasificación de péptidos antimicrobianos es importante contar con la mayor cantidad de secuencias peptídicas con actividad antimicrobiana validada experimentalmente. Para esto se decidió utilizar la base de datos denominada “starPep” (Aguilera-Mendoza *et al.* (2019)), dado que esta es una base de datos que contiene la colección más grande conocida de secuencias de péptidos antimicrobianos. Esta base de datos cuenta con 45,120 secuencias y está organizada por actividad biológica (antibacteriano, antifúngico, anticáncer, antiviral, etc.).

3.1.1. Conjuntos de entrenamiento y validación

Para la generación del modelo de clasificación de una clase, como el que se planteó al inicio del capítulo, fue necesario obtener secuencias que únicamente tuvieran reportado una actividad. Esto sé debe a que al utilizar clasificadores de una sola clase, solo se puede modelar una actividad en específico. Por ejemplo, si nosotros generamos un modelo de clasificación de una clase para detectar péptidos antibacteriales, es ideal que este modelo solo haya sido entrenado con péptidos antibacteriales. Si en el caso anterior, introdujéramos péptidos que tienen tanto actividad antibacterial como antiparasitaria, correríamos el riesgo de que el modelo fuera capaz de reconocer péptidos que únicamente tienen actividad antiparasitaria, por lo tanto, realizando una clasificación incorrecta. Por lo tanto, en la construcción de los conjuntos antibacterial, Gram positivo y Gram negativo se utilizaron secuencias que únicamente tenían reportada esa actividad. Para el caso del conjunto AMP no fue necesario utilizar secuencias con actividad única, dado que por definición este conjunto contiene cualquier actividad que se considere antimicrobiana. A continuación se describe la serie de pasos seguidos para generar los conjuntos utilizados en esta metodología.

Para la construcción de los conjuntos de entrenamiento se recuperaron de la base de datos “starPep” 20,728 secuencias, las longitudes de estas secuencias se encuentran en un intervalo de 3 a 100 aminoácidos; se eliminaron todas aquellas secuencias cuya longitud era mayor a 30 aminoácidos y menor a 10 aminoácidos, dejando un total de 9,868 secuencias únicas. Del total de 9,868 secuencias: 8,036 poseen actividad antibacteriana, 2,640 poseen actividad antifúngica, 2,184 poseen actividad antiviral y 304 poseen actividad antiparasitaria. En la Figura 14 se muestra un diagrama de Venn con el tamaño de la intersección entre cada uno de los conjuntos mencionados, obtenida con el software DoverAnalyzer 0.1.2 (Aguilera-Mendoza *et al.* (2015)).

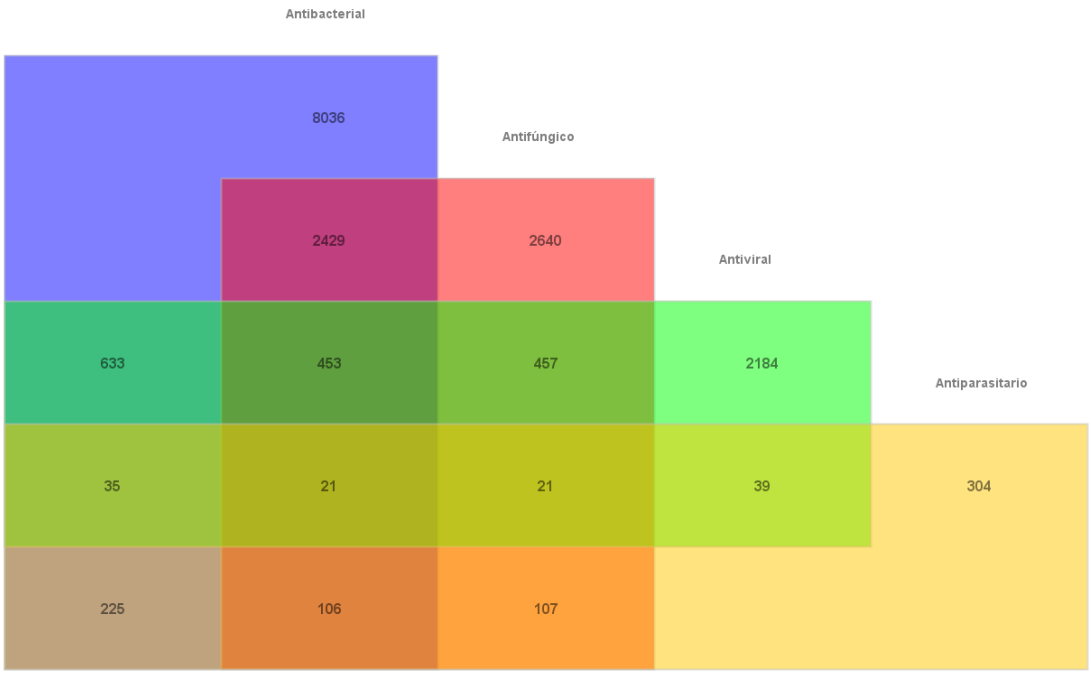


Figura 14. Diagrama de Venn de los subconjuntos que conforman el conjunto de entrenamiento AMP. Este conjunto contiene actividades: antibacterial, antifúngico, antiviral y antiparasitario.

Las secuencias seleccionadas se utilizaron para generar los conjuntos: “AMP”y “Antibacterial”. Para generar el conjunto AMP se utilizaron 6,408 de las 8803 secuencias antibacteriales, así como la totalidad de las secuencias con actividades antifúngicas, antivirales y antiparasitarias, sumando un total de 8803 secuencias únicas después de remover las secuencias duplicadas. Las secuencias utilizadas para crear el conjunto AMP pueden ser encontradas en el archivo “AMP_TR.fasta” del Anexo A. El conjunto “Antibacterial” se generó a partir de las 8,036 secuencias con actividad antibacterial, de estas secuencias se utilizaron únicamente aquellas que en la base de datos “starPep” sus metadatos señalaban que estas poseían únicamente actividad antibacterial.

El tamaño final del conjunto es de 4,844 secuencias únicas, las secuencias se pueden encontrar en el archivo “Antibacterial.fasta”.

Para la generación de los conjuntos Gram positivo y Gram negativo se obtuvieron dos conjuntos de secuencias de “starPep”. El primero, contenía todas las secuencias que poseían actividad Gram negativa con longitud entre los 10 y los 30 aminoácidos, recuperando un total de 6,572 secuencias. El segundo, se obtuvo de manera análoga al primero seleccionando la actividad Gram positiva de la base de datos, recuperando un total de 6,325 secuencias.

Posteriormente, se tomaron de los conjuntos aquellas secuencias que poseían únicamente una de las dos actividades, dando como resultado dos conjuntos: un conjunto Gram negativo, que tiene un tamaño de 716 secuencias, y un conjunto Gram positivo que tiene un tamaño de 469 secuencias; ambos conjuntos se separaron en una proporción 80/20 utilizando el 80 % de las secuencias para entrenamiento y el 20 % para validación. También se obtuvo un conjunto externo obteniendo la intersección de los conjuntos Gram positivo (6,325) y Gram negativo (6,572) y eliminando las secuencias que se utilizaron para entrenamiento y validación. Los conjuntos pueden ser encontrados en el Anexo A con los nombres: "Gram_neg_TR.fasta" (entrenamiento Gram negativo), "Gram_neg_TS.fasta" (validación Gram negativo), "Gram_pos_TR.fasta" (entrenamiento Gram positivo), “Gram_pos_TS.fasta” (validación Gram positivo) y “Gram_EXT_TS.fasta” (validación externa Gram positivo y Gram negativo). En la Figura 15 se muestra la intersección existente entre los conjuntos de entrenamiento mencionados. En la Figura 16 se muestra la partición de secuencias como se describe en esta sección.

3.1.2. Conjuntos negativos de validación

En la validación del modelo se utilizaron dos conjuntos negativos, un conjunto de 1695 secuencias y otro de 10771 secuencias, ambos obtenidos del trabajo presentado por Pinacho-Castellanos *et al.* (2021a). Los conjuntos pueden ser encontrados en el Anexo A, con los nombres “Neg_TS.fasta”, “Neg_EXT_TS.fasta”, respectivamente. Aunque este conjunto no tiene validación experimental, se seleccionó utilizando la metodología mencionada por Xiao *et al.* (2013) y Veltri *et al.* (2018) donde se siguieron los siguientes pasos:

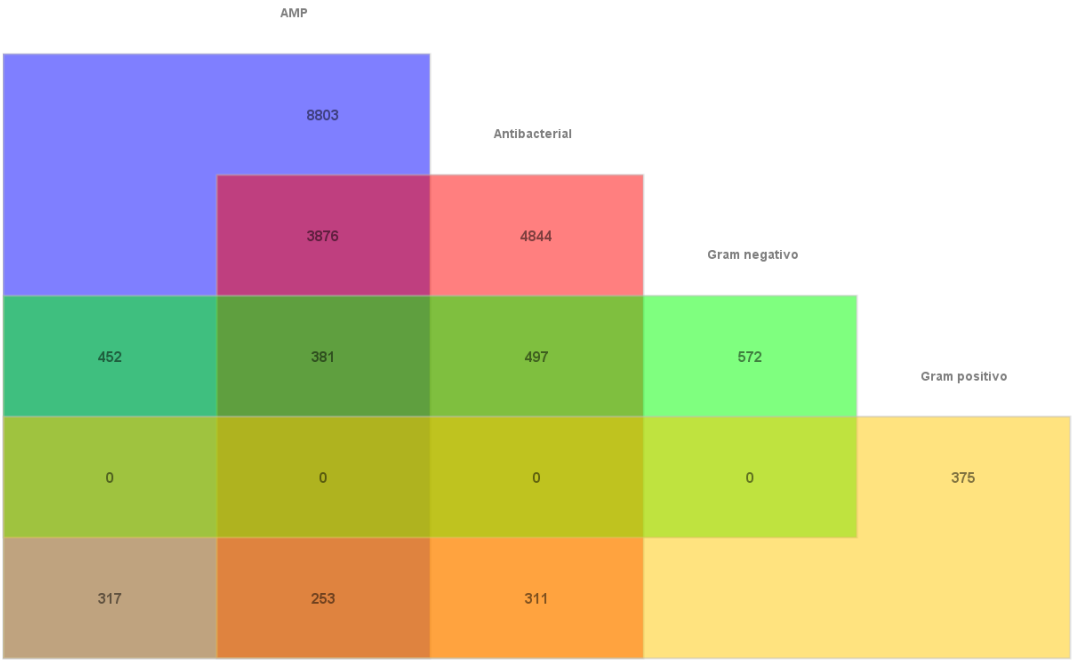


Figura 15. Diagrama de Venn de los conjuntos de entrenamiento. Se muestra el número de secuencias compartidas entre los conjuntos de entrenamiento. El número de secuencias compartidas entre el conjunto Gram positivo y Gram negativo es cero, debido a que se eligieron secuencias con una sola actividad.

1. De la base de datos Uniprot (Bateman, 2019) se seleccionan secuencias con el siguiente criterio de búsqueda: “no antimicrobiano”, “no antibacterial”, “no antibiótica”, “no citoplasma” y “no excretorias”.

2. Se removieron todas aquellas secuencias con longitud menor a 10 aminoácidos y mayor a 100 aminoácidos.

3. Para reducir el sesgo de homología y la redundancia, se utilizó el programa CD-HIT (Li y Godzik, 2006) para eliminar aquellas secuencias con más de 40 % de identidad de secuencia por pares con cualquier otra.

4. Se muestrea de manera aleatoria las secuencias resultantes para generar conjuntos de validación

Los criterios de búsqueda del paso 1 se utilizan debido a que se cree que estos péptidos tienen una baja posibilidad de ser antimicrobianos. Por ejemplo, los péptidos antimicrobianos se secretan al medio extra celular, por lo tanto se espera que los péptidos no secretorios a su vez no posean actividad antimicrobiana. Es importante resaltar que ninguno de los conjuntos de validación se utilizó para entrenar alguno de

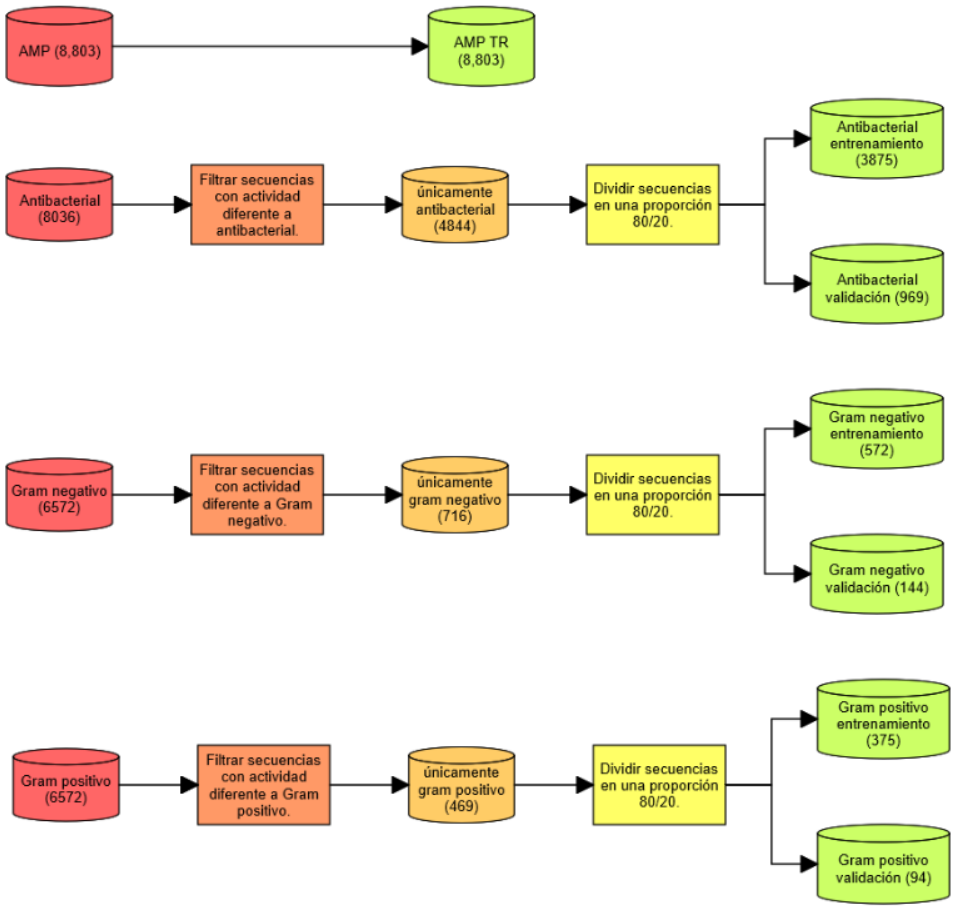


Figura 16. Partición de secuencias en conjuntos. El diagrama muestra los pasos seguidos para generar los conjuntos de entrenamiento y de prueba.

los modelos; así como tampoco se utilizó para generar algún proceso de optimización de hiperparámetros basado en estos, dado que esto introduciría el sesgo que se trata de eliminar utilizando modelos de una clase.

3.2. Descriptores moleculares

En el cálculo de descriptores moleculares de los modelos generados se utilizaron las herramientas de starPep (Aguilera-Mendoza *et al.* (2019)), iFeature (Chen *et al.* (2018)), ProtDcal (Romero-Molina *et al.* (2019)) y BERT ESM (Rives *et al.* (2019)). Se decidió utilizar estas herramientas principalmente por su capacidad de generar múltiples descriptores basados en diferentes algoritmos y propiedades fisicoquímicas de los aminoácidos.

Tabla 4. Parámetros seleccionados para la generación de descriptores moleculares en starPep. Se enlistan todos los parámetros que starPep utiliza para generar descriptores moleculares y se marca con un "✓" aquellos utilizados.

Operadores de agregación		**Propiedades químicas**		**Fragmentos químicos**	
Operador	**Valor**	**Propiedades**	**Valor**	**Fragmentos**	**Valor**
N1 - Manhattan Norm	✓	Relative reverse-trun frequency	✓	Total	✓
N2 - Euclidean Norm	✓	Geometric computability parameter-1	✓	Aliphatic	✓
AM - Arithmetic Mean	✓	Geometric computability parameter-2	✓	Favoring Alpha Helix	✓
P2 - Quadratic Mean	✓	Heat of formation	✓	Apolar	✓
P3 - Potential Mean	✓	Side chain mass	✓	Aromatic	✓
HM - Harmonic Mean	✓	Side chain volume	✓	Favoring Beta Sheet	✓
V - Variance	✓	Isoelectric point	✓	Favoring Beta Turn	✓
S - Skewness	✓	Relative alpha-helix frequency	✓	Negatively Charged Polar	✓
K - Kurtosis	✓	Relative beta-sheet frequency	✓	Positively Charged Polar	✓
SD - Standard Deviation	✓	Isotropic surface area	✓	Uncharged Polar	✓
VC - Variation Coefficient	✓	Z1-scale	✓	Unfolding	✓
RA- Range	✓	Z2-scale	✓	**Operadores de agregación clasicos**	
i50 - Inter-percentil difference	✓	Z3-scale	✓	**Operador**	**Valor**
GOWAWA	✗	Boman	✓	K value	3
CHOQUET - Choquet Integral	✗	Charge	✓	AC- Autocorrelation	✓
		Hidrophilicity	✓	GV - Gravitational	✗
				TS - Total Sum	✗
				ES - Electro-Topological State	✓

3.2.1. Generación de descriptores moleculares de starPep

Para la generación de descriptores moleculares con starPep se utilizó la versión 0.8.5 del software. Se decidió utilizar todas las características disponibles en el apartado de índices basados en operadores de agregación. Este apartado puede ser encontrado en starPep en la siguiente ruta: "Tools", "Extraction", "Extended indices", "Indices based on aggregation operators". En la Tabla 4 se presentan las características disponibles en este apartado de starPep, así como la selección utilizada para la generación de descriptores.

Este proceso de descripción molecular genera alrededor de 10,000 descriptores moleculares, dependiendo del tamaño del conjunto analizado. Se generaron dos modelos basados en estos descriptores, uno seleccionando características con el algoritmo presentado en la sección 3.3, utilizando el filtro de correlación de Spearman al 90 % y el segundo utilizando este filtro al 95 %.

3.2.2. Generación de descriptores moleculares de ProtDcal

Para la generación de conjuntos basados en ProtDcal, se utilizaron los conjuntos obtenidos por Pinacho-Castellanos *et al.* (2021b), dado que estos ya habían probado dar buenos resultados en su aplicación. Se tomaron dos conjuntos de los generados por

Tabla 5. Características seleccionadas por Pinacho-Castellanos *et al.* (2021b) para el modelo AMP. Se enlistan los índices, operadores de vecindad, grupos y operadores de agregación que se utilizaron para generar los descriptores del conjunto AMP.

Indices	Operadores de vecindad	Grupos	Operadores de agregación
ISA, Xi, Gs(U), Mw, L1-9, Z1 , Z3 , HP , DHf', Gw(U), ECI, W(U), Pb, Pt, IP, Pa, Z2, Ap	ES, AC4, AC5, AC2, AC3, KH3, AC1, NO	AHR, PLR, BSR, NPR, RTR, ALR, ARM, PCR, MET, TYR,UCR, PRO, LYS, HIS, CYS, ASN, THR, TRP, ILE, UFR, LEU, GLN, NCR, GLU	I50, N3, Q2, G, Ar, P3, N1, N2, RA, Q1, MN, M, P2, V, SI50, MX, Q3, CV, S, DE

Tabla 6. Características seleccionadas por Pinacho-Castellanos *et al.* (2021b) para el modelo antibacterial. Se enlistan los índices, operadores de vecindad, grupos y operadores de agregación que se utilizaron para generar los descriptores del conjunto antibacterial.

Indices	Operadores de vecindad	Grupos	Operadores de agregación
ISA, Xi, Z3, DHf, Z1, IP, Z2, HP, Mw, W(U), ECI, Gw(U), Pt, Gs(U), Ap, L1-9,Pa, Pb	ES, AC4, KH3, AC1, AC3, AC2, AC5, NO	AHR, PLR, NPR, BSR, ALR, ARM, PCR, UFR, RTR, NCR, MET, CYS, UCR, TRP, LYS, TYR, ASN, THR, GLN, ILE	I50, Q2, P3, RA, P2, DE, G, N1, M, MN, Q3, V, Q1, N2, SI50, Ar, MX, N3

Pinacho-Castellanos *et al.* (2021b), el conjunto de características AMP y conjunto de características usado para el modelo antibacterial. El conjunto de características AMP se utilizó únicamente en la etapa AMP del modelo ProtDcal; el modelo antibacterial se utilizó para las etapas antibacterial, Gram positiva y Gram negativa. En las tablas 5 y 6 se muestran las características de ProtDcal seleccionadas por Pinacho-Castellanos *et al.* (2021b) para los modelos AMP y antibacterial, respectivamente.

Estos conjuntos de descriptores no se sometieron a ningún proceso de selección de características debido a que en Pinacho-Castellanos *et al.* (2021b) se menciona que estos ya pasaron por un proceso de selección. En el proceso de selección utilizado por Pinacho-Castellanos *et al.* (2021a): primero, se separaron los descriptores en conjuntos con base en su índice. Después, a estos conjuntos de descriptores se le aplicó un filtro basado en entropía en el cual se seleccionó las primeras 40 características de cada conjunto. Por último, se utilizó la técnica de selección de características de envoltura, implementada en WEKA versión 3.8.4, con el nombre "WrapperSubsetEval", utilizando el método de búsqueda "GeneticSearch" y ajustando el parámetro "classifier" a "RandomForest". Los 207 descriptores por Pinacho-Castellanos *et al.* (2021a) para la etapa AMP, se utilizaron en la misma etapa de nuestro modelo, y el conjunto de 93 descriptores de la etapa antibacterial se utilizaron en las etapas antibacterial, Gram positivo y Gram negativo de nuestro modelo.

Tabla 7. Lista de algoritmos utilizados en iFeature para la generación de descriptores moleculares. Se enlistan los algoritmos seleccionados del software iFeature, así como los parámetros que se utilizaron en cada algoritmo.

Algoritmo	Parámetros
Composition of K-Spaced Amino Acid Pairs (CKSAAP)	$K_Space = 4$,
Amino acid Composition (AAC)	N/A
Dipeptide Composition (DPC)	N/A
Dipeptide Deviation drom Expected Mean (DDE)	N/A
Tripeptide Composition (TPC)	N/A
Grouped Amino Acid Composition (GAAC)	N/A
Composition of K-Spaced Amino Acid Group Pairs (CKSAAGP)	N/A
Grouped Dipeptide Composition (GDPC)	N/A
Grouped Tripeptide Composition (GTPC)	N/A
Normalized Moreau-Broto Autocorrelation	AAindex=ANDN920101;ARGP820101;ARGP820102;ARGP820103; BEGF750101;BEGF750102;BEGF750103;BHAR880101 $Lag_Value = 3$
CTD Composition	N/A
CTD Transition	N/A
CTD Distribution	N/A
Conjoint Triad	N/A
K-Spaced Conjoint Triad	$K_Space = 4$
Sequence-Order-Coupling Number	$Lag_Value = 3$
Quasi-Sequence-Order	$Lag_Value = 3$
Pseudo-Amino Acid Composition	$Weight_Value = 0.1$ $Lamada_Value = 3$
Amphiphilic Pseudo-Amino Acid Composition	$Weight_Value = 0.1$ $Lamada_Value = 3$

3.2.3. Generación de descriptores moleculares de iFeature

De la librería de descriptores moleculares para proteínas de iFeature, se utilizaron únicamente aquellos que se pueden utilizar en conjuntos de péptidos con distintas longitudes. Esta librería contiene una variedad de algoritmos para el cálculo de descriptores que han sido recopilados de la literatura e implementados en el lenguaje Python 3 (Van Rossum y Drake, 2009). En la Tabla 7 se presenta la lista de algoritmos utilizados y los hiperparámetros que fueron utilizados en cada algoritmo.

Este proceso para el cálculo de descriptores permite generar 11,770 descriptores independientemente del tamaño del conjunto o de la composición de las secuencias. Esta representación se optimizó utilizando el algoritmo de la sección 3.3, utilizando un umbral de correlación de Spearman de 90 %.

3.2.4. Representación por codificación de BERT-ESM

En el proyecto de BERT-ESM (Devlin *et al.*, 2019) se incluye una serie de redes neuronales entrenadas con diferentes conjuntos de entrenamientos y generadas utilizando distintas arquitecturas. El modelo que fue utilizado para generar la representación

es el modelo ESM-1, este modelo utiliza 650 millones de parámetros optimizables y su arquitectura posee 33 capas (niveles) de codificación (ver Figura 17). Para generar la representación de esta red neuronal, es necesario introducir únicamente los parámetros: I) el número/etiqueta de las capas que se van a utilizar para generar la representación; II) El tipo de representación: “per_tok”, “mean” y “bos” (ver sección 2.3.4). Para la generación de la representación se utilizó únicamente la información de la capa 33, en conjunto con el parámetro “mean”.

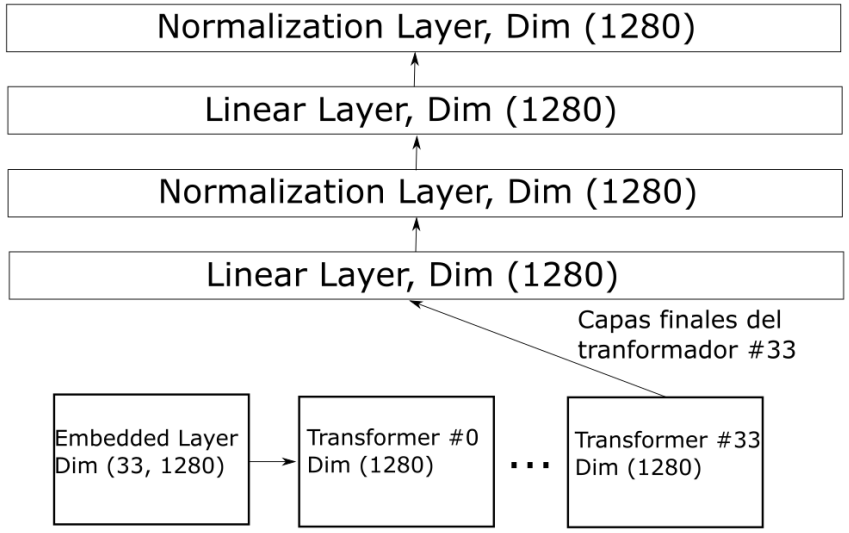


Figura 17. Diagrama de la arquitectura BERT-ESM. La representación utilizada, es la obtenida partir de la última capa de normalización del transformador 33.

La red neuronal tiene una representación fija de 1280 valores para cada péptido que se desee codificar. Con esta representación se generaron dos modelos. El primero, se generó utilizando la representación completa de la red neuronal (1280 valores). El segundo, se generó utilizando el algoritmo de selección de características de la sección 3.3, utilizando un umbral de correlación de Spearman de 90 %.

3.3. Selección de características

Para el proceso de selección de características se utilizó el algoritmo presentado por Aguilera-Mendoza *et al.* (2020), que además es el que utiliza starPep para el mismo proceso. Aunque el algoritmo es no supervisado, se deben introducir manualmente dos parámetros, un umbral para el filtro de entropía de Shannon, y seleccionar entre utilizar la correlación de Spearman o la de Pearson e introducir un umbral para este

filtro también. El proceso de selección de característica se divide en dos etapas: el filtro de descriptores y la optimización de subconjunto.

3.3.1. Primera Etapa

En la primera fase, se aplican filtros basados en entropía de Shannon y correlación de Spearman o Pearson. Para esto se siguen los siguientes pasos:

1. Se calcula el valor de entropía de Shannon para cada uno de los descriptores disponibles.
2. Los descriptores moleculares se ordenan en forma descendente con relación a su entropía y se eliminan los peores descriptores con base en su posición en el ordenamiento. Las muestras eliminadas son todas aquellas cuyo valor de entropía está por debajo del umbral seleccionado.
3. Se calcula el valor de la correlación de Spearman para cada par de descriptores.
4. Se genera una matriz con los valores de correlación entre cada para de descriptores. De aquellos descriptores cuyo porcentaje de correlación sea mayor al umbral especificado, se eliminara el que tenga el menor valor de entropía.

3.3.2. Segunda Etapa

Utilizando los descriptores moleculares que no fueron eliminados por la primera etapa de filtrado de características, se inicia un proceso de optimización de subconjunto donde se busca un subconjunto de descriptores que mejor satisfaga los parámetros de la siguiente ecuación:

$$\max_{F \epsilon \Omega} \quad \phi(F) = \frac{1}{|F|} \sum_{f_j \epsilon F} H(f_j) - \frac{1}{|F|^2} \sum_{f_j, f_k \epsilon F} I(f_j, f_k) \tag{18}$$

donde $\phi(F)$ es la función objetivo, y F es un subconjunto de características sobre el espacio de búsqueda de todos los posibles subconjuntos del conjunto de características.

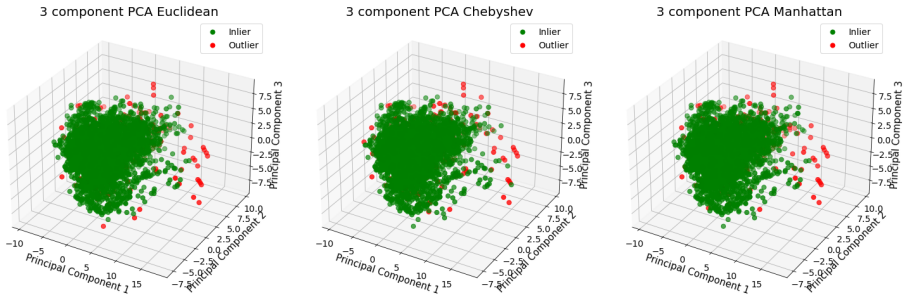
La función $H(f_j)$ representan los valores de entropía de los descriptores y la función $I(f_j, f_k)$ los valores de la correlación de Spearman para los descriptores f_j y f_k.

Inclusive después de la etapa de filtrado de descriptores, es posible que la cantidad de descriptores restantes para la etapa de optimización de subconjuntos sea tal que el proceso de optimización se vuelva computacionalmente costoso. Para evitar este alto costo computacional, Aguilera-Mendoza *et al.* (2020) proponen la utilización de una heurística del tipo "Greedy Hill-Climber". Al utilizar esta heurística, se comienza con un conjunto que contiene todos los descriptores moleculares que no fueron eliminados en la primera etapa. Estos descriptores se eliminan de manera progresiva, buscando maximizar el resultado de la ecuación 18. La condición de paro, se presenta cuando el algoritmo es incapaz de encontrar una eliminación que contribuya a incrementar el valor de $\phi(F)$. Al final de este proceso, los descriptores resultantes deben ser aquellos que presenten los valores de entropía más alta con el menor porcentaje de correlación entre ellos. Es importante mencionar que esta metodología no garantiza la obtención de un conjunto óptimo; sin embargo, se considera que produce conjuntos suficientemente buenos.

3.4. Filtro basado en Local Outlier Factor

Con el propósito de filtrar las muestras de los conjuntos de entrenamiento que tengan la mayor probabilidad de ser anomalías, se utilizó un filtro de secuencias basado en "Local Outlier Factor" utilizando las distancias: "Euclidean", "Chebyshev" y "Manhattan". El algoritmo de "Local Outlier Factor" no se utilizó para eliminar las muestras anómalas, sino que este se utilizó a modo de puntuación; las muestras anómalas se eliminaron siguiendo el criterio de la "Boxplot Outlier Rule" (Hoaglin *et al.* (1986)). A continuación se presentan los pasos que se siguieron para eliminar anomalías:

1. Se calcula el Local Outlier Factor para cada una de las secuencias en el conjunto analizado utilizando la distancia Euclidean.
2. Se calcula el rango intercuartílico de la distribución generada por el paso 1 además de la distancia intercuartílica.



(a) (b) (c)

Figura 18. Filtro de anomalías utilizando el algoritmo Local Outlier Factor con distancias: (a) Euclidean (b) Chebyshev (c) Manhattan. En cada una de las imágenes se muestra las instancias que son filtradas por el algoritmo de detección de anomalías en color rojo y en color verde las muestras no anómalas.

3. Se consideran como anomalías todas las muestras cuyo valor de “Local Outlier Factor” sea: i) mayor al valor de “Q75” más 1.5 de la distancia intercuartílica; ii) menor al valor de “Q25” menos 1.5 de la distancia intercuartílica.

4. Se repiten los pasos del 1 al 3 utilizando las distancias “Chebyshev” y “Manhattan” para calcular el “Local Outlier Factor”.

El filtro anteriormente mencionado se utilizó sobre todos los conjuntos de entrenamiento mencionados en 3.1.1 (AMP, antibacterial, Gram positivo, Gram negativo). Después de aplicar este procedimiento a los conjuntos de entrenamiento se generan tres subconjuntos diferentes, uno por cada una de las distancias utilizadas. Durante la experimentación presentada estos conjuntos no se mezclaron entre sí. Es decir, aquellos modelos entrenados con conjuntos de entrenamiento filtrados utilizando distancia euclidiana se utilizaron únicamente en conjunto con aquellos modelos filtrados con distancia euclidiana.

En la Figura 18 se puede observar un ejemplo donde se utilizó la librería de PCA de scikit-learn 0.24.2 (Pedregosa *et al.* (2011)) y Matplotlib (Barrett *et al.* (2005)) para visualizar el filtrado del conjunto antibacterial mencionado en 2.1 utilizando las tres distancias mencionadas “Euclidean”, “Chebyshev”, “Manhattan”.

3.5. Modelos

Para la generación de modelos de detección de anomalías (ver sección 2.6.2) se utilizó el software Weka (Hall *et al.* (2009)) en su versión 3.8.0. Se utilizaron tres tipos de modelo: un ensamble de algoritmos de detección de anomalías que utiliza el método "bagging" y cuyas instancias individuales son del tipo "Isolation Forest" (Liu *et al.* (2008)). También, un ensamble de algoritmos de detección de anomalías que utiliza el método "bagging" y cuyas instancias individuales son del tipo "Local Outlier Factor" (Breuniq *et al.* (2000)). Por último, se utilizó un clasificador de una sola clase (ver en sección 2.6.) conformado por un ensamble de algoritmos de detección de anomalías que utiliza el método "bagging" y cuyas instancias individuales son del tipo "BayesNet" (Heckerman, 1997). Las complejidades de estos algoritmos son: $O(nlogn)$ para el caso del "Isolation Forest" y $O(n^2)$ para el caso de los algoritmos "Local Ouliter Factor" y "BayesNet".

Se decidió utilizar estos tres algoritmos principalmente porque han sido ampliamente utilizados en la literatura; por ejemplo, en los casos presentados por McLachlan *et al.* (2020), Alsini *et al.* (2021) y Vijayakumar *et al.* (2020). Otro factor importante por el cual se decidió utilizar estos algoritmos es porque en su implementación en Weka, no fue necesario modificar hiperparámetros para que estos presentaran buenos resultados. Esto, a diferencia de su implementación en scikit-learn 0.24.2, donde sí se presentó este problema.

Para cada uno de los conjuntos de entrenamiento mencionados en la sección 3.1.1 (AMP, antibacterial, Gram positivo, Gram negativo) se generaron los tres modelos mencionados en esta sección generando un total de doce modelos de detección de anomalías, uno por cada uno de los filtros basados en "Local Outlier Factor" como se menciona en la sección 3.4. Estos modelos no se implementaron de manera individual, sino que se utilizaron como se explicará en la sección 3.7.

Todos los modelos de aprendizaje máquina dependen de una serie de hiperparámetros para sintonizar los modelos, estos son valores que cada algoritmo necesita para operar y que deben ser definidos por el usuario antes de entrenar el modelo. Algo común en los modelos de aprendizaje de máquina es que estos sean sensibles a los cambios de hiperparámetros utilizados en la generación del modelo; en muchos ejem-

plos de la literatura se utilizan algoritmos de optimización para encontrar los mejores hiperparámetros para cada modelo utilizado para un problema determinado.

Para los algoritmos utilizados en esta implementación, existen hiperparámetros que poseen una mayor influencia en la definición de los límites de decisión del modelo. Para el caso del algoritmo “Isolation Forest”, el parámetro determinante es el número de árboles (“numTrees”). En el caso del algoritmo “Local Outlier Factor” son cotas de número de vecinos a tomar en consideración (“minPoints Lower Bound”, “minPoints Upper Bound”). En el algoritmo “BayesNet” no existe un parámetro escalar a optimizar, sino que se pueden probar diferentes métodos de estimación (“estimator”) y algoritmos de búsqueda (“searchAlgorithm”).

Comúnmente, se busca una optimización de hiperparámetros que beneficie a la precisión total del modelo, es decir, que obtenga los mejores resultados posibles al clasificar tanto las muestras positivas como las muestras negativas. En nuestro caso, al contar únicamente con la validación del conjunto positivo, solo nos sería posible optimizar los hiperparámetros para mejorar la recuperación de este conjunto. La razón por la cual sé decidió no realizar este proceso fue debido a que durante la experimentación se notó que al utilizar los valores que Weka utiliza por omisión, ya se obtenía una alta recuperación de las muestras positivas por parte de los modelos. Además, maximizar únicamente la capacidad de recuperación de la clase positiva, podría afectar a la capacidad de generalización del modelo. Por lo tanto, se tomó la decisión de utilizar los parámetros por omisión de Weka. Los hiperparámetros utilizados para cada modelo se muestran en las tablas 8, 9 y 10.

Tabla 8. Parámetros para el Bagging Isolation Forest. Se enlistan los parámetros utilizados por el software Weka para el algoritmo Isolation Forest, así como también se enlistan los valores utilizados.

Parámetros	
Parámetro	**valor utilizado**
batchSize	100
debug	False
doNotCheck Capabilities	False
numDecimal Places	2
numTrees	100
seed	1
subsampleSize	256

Tabla 9. Parámetros para el Bagging Local Outlier Factor. Se enlistan los parámetros utilizados por el software Weka para el algoritmo Local Outlier Factor, así como también se enlistan los valores utilizados.

Parámetros	
Parámetro	**valor utilizado**
batchSize	100
debug	False
doNotCheck Capabilities	False
minPoints LowerBound	10
minPoints UpperBound	40
numDecimal Places	2
num Execution Slots	1

Tabla 10. Parámetros para el OneClass Bagging Bayes Net. Se enlistan los parámetros utilizados por el software Weka para el algoritmo BayesNet, así como también se enlistan los valores utilizados.

Parámetros	
Parámetro	**valor utilizado**
NNSearch	Euclidean Distance
batchSize	100
debug	False
doNotCheck Capabilities	False
estimator	SimpleEstimator
searchAlgorithm	K2
useADTree	False

3.6. Clustering

Al final del proceso de descripción molecular, las moléculas representadas por las características escogidas por los algoritmos de optimización de subconjunto (Shannon y Spearman) forman un espacio euclidiano multidimensional conocido como espacio químico.

En la Figura 18 se observa un cúmulo de puntos, en el cual cada punto representa un péptido definido por el espacio de descriptores; esta representación gráfica es una proyección de un espacio químico, el cual está definido por 98 descriptores moleculares a un espacio de tres dimensiones. La Figura 19 fue obtenida utilizando la herramienta Glueviz 0.15.2 (Robitaille *et al.* (2019)), gracias a que los algoritmos de proyección mantienen las distancias entre cada par de puntos (PCA), esta visualiza-

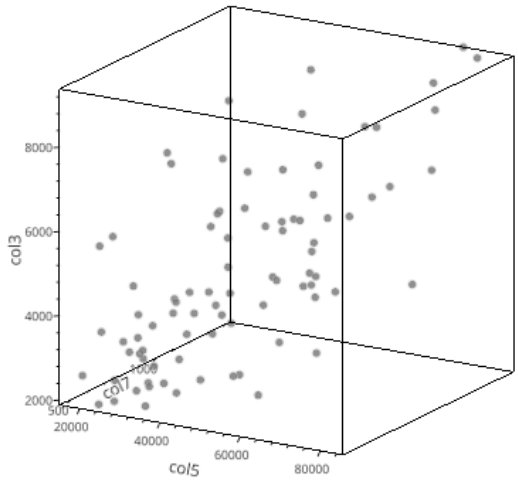


Figura 19. Representación de un conjunto de péptidos antibacteriales. Se utilizó el software Glueviz para representar el conjunto antibacterial. Los puntos grises representan una instancia del conjunto.

ción nos permitió apreciar regiones del espacio donde se pueden encontrar una mayor densidad de péptidos.

Es de nuestro interés encontrar agrupaciones de puntos o “comunidades” en este espacio de características debido a que esto puede indicar que existen similitudes entre determinadas secuencias que van más allá de la actividad, además que nos permite hacer una mejor clasificación de nuevas muestras. Para encontrar estas agrupaciones de puntos se utilizaron los algoritmos de “clustering”: “Expectation maximization” y “K-means”. Los algoritmos se utilizaron con todos los conjuntos de entrenamiento mencionados en la sección 3.1.1. El procedimiento fue el siguiente: primero, se utilizó el algoritmo “Expectation maximization” sobre el conjunto objetivo, esto debido a que este algoritmo no necesita que el usuario introduzca el número de clusters a construir. Después, se utiliza sobre el mismo conjunto el algoritmo “K-means” utilizando las distancias “Euclidean” y “Manhattan” para calcular la distancia entre los puntos y se utiliza como número de “clusters” la cantidad encontrada por “Expectation maximization”. Al final de este proceso obtenemos tres conjuntos de “cluster” distintos, aquellos calculados utilizando “Expectation maximization”, aquellos calculados utilizando “K-means” con distancia “Euclidean” y aquellos calculados utilizando “K-means” con distancia “Manhattan”. Esta combinación de distancias se realiza para analizar si alguna de ellas permite lograr un mejor desempeño de los algoritmos.

3.7. Proceso jerárquico de detección de anomalías y validación

Como se mencionó, esta metodología tiene como objetivo determinar si una secuencia de péptido tiene o no propiedades antibacterianas Gram negativas, Gram positivas o ambas. Para esto utilizamos una serie de filtros: primero, se utiliza un método de detección de anomalías entrenado con el conjunto AMP mencionado en 3.1.1 y que tiene la tarea de decidir si una muestra tiene o no propiedades antimicrobianas. Después, se utiliza un método de detección de anomalías entrenado con el conjunto antibacterial mencionado en 3.1.1 y que decide si la muestra tiene o no propiedades antibacterianas. Por último, si se decide que la muestra tiene propiedad Gram positiva o Gram negativa, se somete la muestra a dos métodos de detección de anomalías, uno entrenado con muestras Gram negativas, otro entrenado con Gram positivas y se decide si la muestra pertenece a alguno de estos dos conjuntos o ambos. Durante el proceso de clasificación de las muestras puede existir el caso donde una muestra pertenezca a ambos conjuntos (Gram positivo y Gram negativo), esto indica que la muestra analizada posee ambas actividades dado que estas no son mutuamente excluyentes. A continuación, se explican los pasos para implementar esta metodología, la cual se ilustra en la Figura 20:

1. Seleccionar las muestras que se quieren probar y someterlas a un proceso de descripción molecular donde se seleccionen los descriptores utilizados en el conjunto AMP.

2. Utilizar el modelo “Oneclass Bagging BayesNet” entrenado con el conjunto AMP para decidir si las muestras analizadas pertenecen o no a este conjunto.

3. Si el modelo anterior decide que una muestra pertenece a su conjunto, se genera un modelo de los mencionados en 3.5 para cada uno de los clusters que se generaron en 3.6.

4. La muestra analizada se prueba en cada uno de los métodos de detección de anomalías entrenados en el paso 3, si alguno de estos modelos predice que la muestra pertenece a su conjunto, entonces, se considera que la muestra pertenece al conjunto AMP. Si el modelo determina que no pertenece al conjunto AMP entonces la muestra se considera una anomalía.

5. Se repiten los pasos del 1 al 4 para los conjuntos antibacterial, Gram positivo y Gram negativo.

3.7.1. Validación

Para la etapa de validación, se utilizan los conjuntos de validación Gram positivo, Gram negativo y el conjunto de prueba negativo. Todas las muestras de estos conjuntos se someten al proceso descrito anteriormente y en la Figura 9. Después, se toman todas la muestras clasificadas correcta e incorrectamente de los conjuntos de prueba y se calculan los parámetros de sensibilidad (recuperación de la clase positiva) y especificidad (recuperación de la clase negativa).

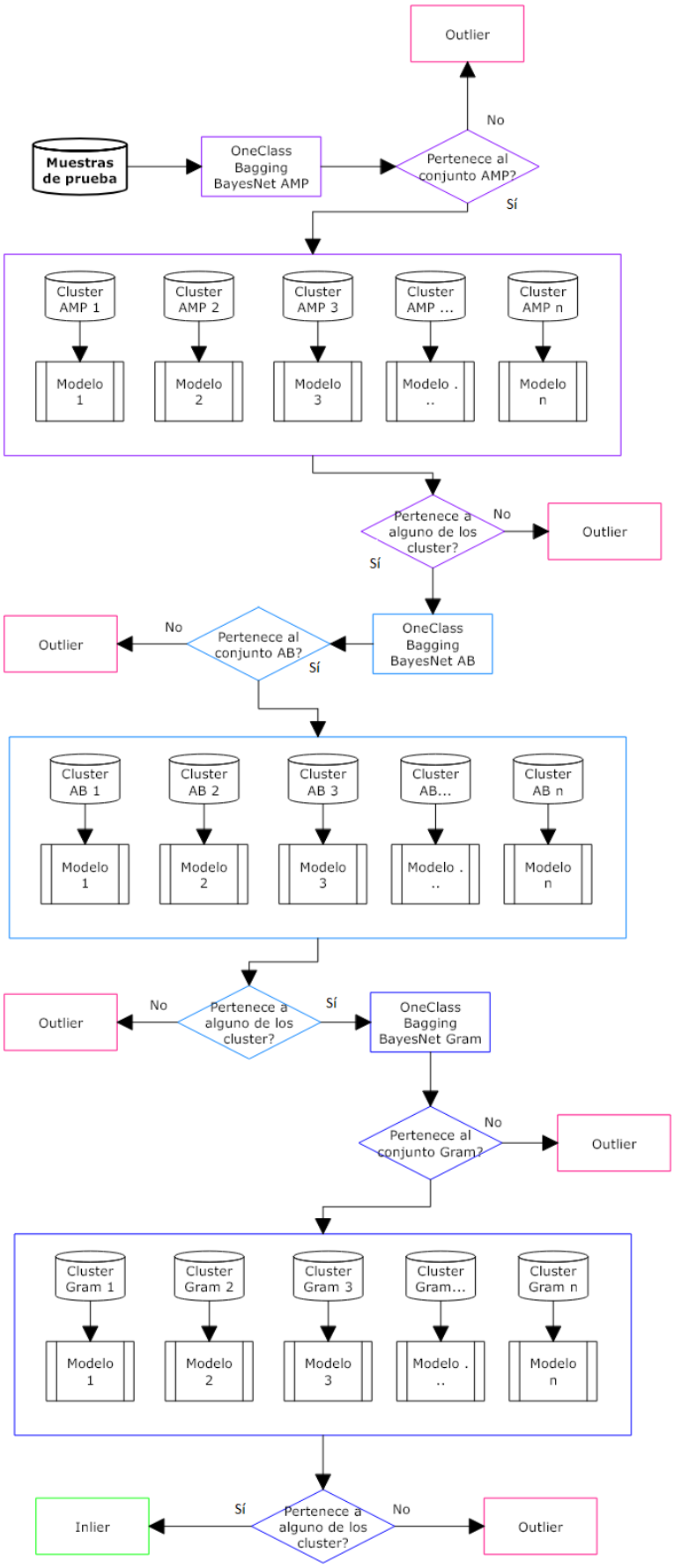


Figura 20. Flujo de clasificación jerárquica. Diagrama del proceso de clasificación jerárquica desde la etapa AMP hasta la etapa Gram.

Capítulo 4. Resultados

En este capítulo se presentan los resultados obtenidos a partir de la utilización de los modelos que se construyeron utilizando la metodología de clasificación de una sola clase presentada en el Capítulo 3. El objetivo de esta sección es demostrar el cumplimiento de los objetivos presentados en el Capítulo 1, así como la comparación de la metodología con el estado del arte.

4.1. Generación de conjuntos de entrenamiento y validación

Para generar un modelo como el que se propone en el Capítulo 3, fue necesario construir nuevos conjuntos de péptidos. Esto debido principalmente a que no se encontraron conjuntos de péptidos Gram negativos y Gram positivos previamente generados en la literatura. Para seleccionar estas secuencias utilizamos la base de datos “starPep”, en donde buscamos la actividad biológica asociada a cada secuencia en la etiqueta “related_to” de los metadatos.

Se construyeron cuatro conjuntos de entrenamiento a partir de las secuencias encontradas. Se generó un conjunto denominado “AMP” (Antimicrobial), el cual contiene secuencias con actividades: antibacterial, antiviral, antiparasitario y antifúngico; este conjunto está conformado por 8803 secuencias. Se generó también un conjunto de entrenamiento denominado “antibacterial” que contiene secuencias cuya etiqueta “realted_to” de los metadatos las asocia únicamente a la actividad antibacterial; este conjunto está conformado por 4844 secuencias. Por último, se construyeron dos conjuntos con actividad antibacterial, en los cuales se incluyeron secuencias cuya etiqueta “related_to” las asocia únicamente a la actividad Gram negativa o Gram positiva, respectivamente. Estos conjuntos tienen un tamaño de: 572 secuencias para el conjunto Gram negativo y de 375 secuencias para el conjunto Gram positivo. Esta distribución se muestra en la Tabla 11.

Para el proceso de validación se utilizaron cinco conjuntos, los cuales se obtuvieron como se describe en el Capítulo 3. De estos conjuntos de validación, tres contienen secuencias con actividad antimicrobiana: Conjunto Gram negativo, el cual consiste en 144 secuencias, las cuales únicamente tienen reportada actividad Gram negativa;

Tabla 11. Conjuntos de entrenamiento. Se enlistan los conjuntos que se utilizaron para entrenar las etapas de detección del modelo jerárquico, así como el número de secuencias que contienen.

Conjunto	No. de secuencias
Amtimicrobial	8803
Antibacterial	4844
Gram Positivo	572
Gram Negativo	375

conjunto Gram positivo, el cual contiene 94 secuencias, las cuales únicamente tienen reportada actividad Gram positiva; conjunto de validación externo el cual contiene 5856 secuencias con actividad tanto Gram positiva como Gram negativa. Los dos conjuntos restantes se obtuvieron utilizando la metodología mencionada en Xiao *et al.* (2013) y Veltri *et al.* (2018), se utilizan como referente de la clase no-antimicrobial; estos conjuntos contienen 1695 y 10771 secuencias, respectivamente. La denominación y distribución de estos conjuntos se muestra en la Tabla 12.

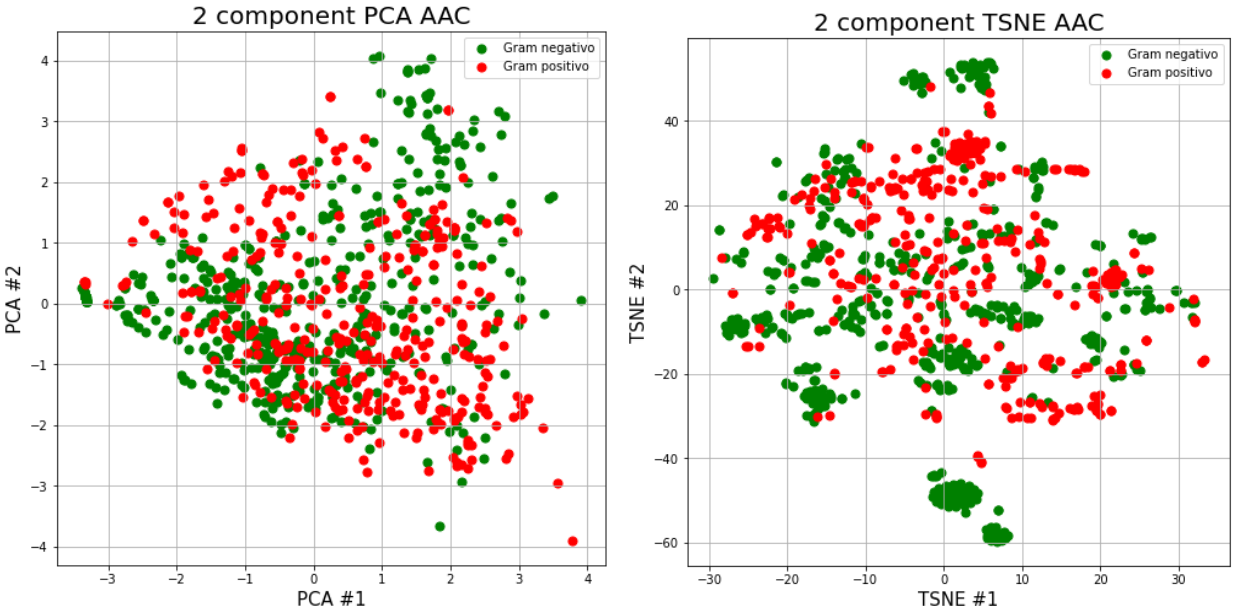
Tabla 12. Conjuntos de prueba. Conjuntos utilizados en la validación de los modelos. El conjunto positivo externo contiene muestras tanto Gram positivas como Gram negativas.

Conjunto	No. de secuencias
Gram negativo	144
Gram positivo	94
Positivo externo	5856
Negativo	1695
Negativo externo	10771

Todos los conjuntos de secuencias mencionadas en esta sección se encuentran en el Anexo A.

4.1.1. Proyección del conjunto de datos utilizando PCA y TSNE

La metodología presentada en el Capítulo 3, se diseñó con el objetivo de poder diferenciar entre péptidos únicamente Gram negativos y péptidos únicamente Gram positivos. Para lograr esto, se entrenó las etapas de clasificación de péptidos Gram negativos y Gram positivos, con secuencias que solamente tenían asociada una de las dos actividades. Por lo tanto, fue de nuestro interés encontrar si es que estos dos conjuntos se diferenciaban una vez que se representan por medio de descriptores moleculares.



(a) (b)

Figura 21. PCA y TSNE de composición de aminoácidos. (a)Representación de los conjuntos de entrenamiento utilizando la composición de aminoácidos y PCA. (b) Representación de los conjuntos de entrenamiento utilizando la composición de aminoácidos y TSNE.

Para esto, se planteó un experimento donde se utilizan los métodos de reducción de dimensionalidad: PCA y TSNE, para obtener una proyección en dos dimensiones de tres de las representaciones utilizadas en los conjuntos de entrenamiento y validación. Las representaciones elegidas fueron: la composición de aminoácidos, los descriptores moleculares de starPep y la representación de la red BERT-ESM. La implementación de los algoritmos PCA y TSNE fue la que se encuentra en la librería sklearn, versión 0.24.1 (Pedregosa *et al.* (2011)).

Las primeras dos proyecciones se obtuvieron a partir de la representación generada por la composición de aminoácidos de los conjuntos Gram negativo y Gram positivo. La composición de aminoácidos es un vector que contiene el número de aminoácidos de cada tipo normalizado con el número total de residuos. Esta representación se eligió para este experimento, debido a que esta nos puede dar una intuición sobre si es que existe algún aminoácido que diferencie el conjunto Gram negativo del Gram positivo y viceversa. Las proyecciones PCA y TSNE de esta representación se pueden observar en la Figura 21.

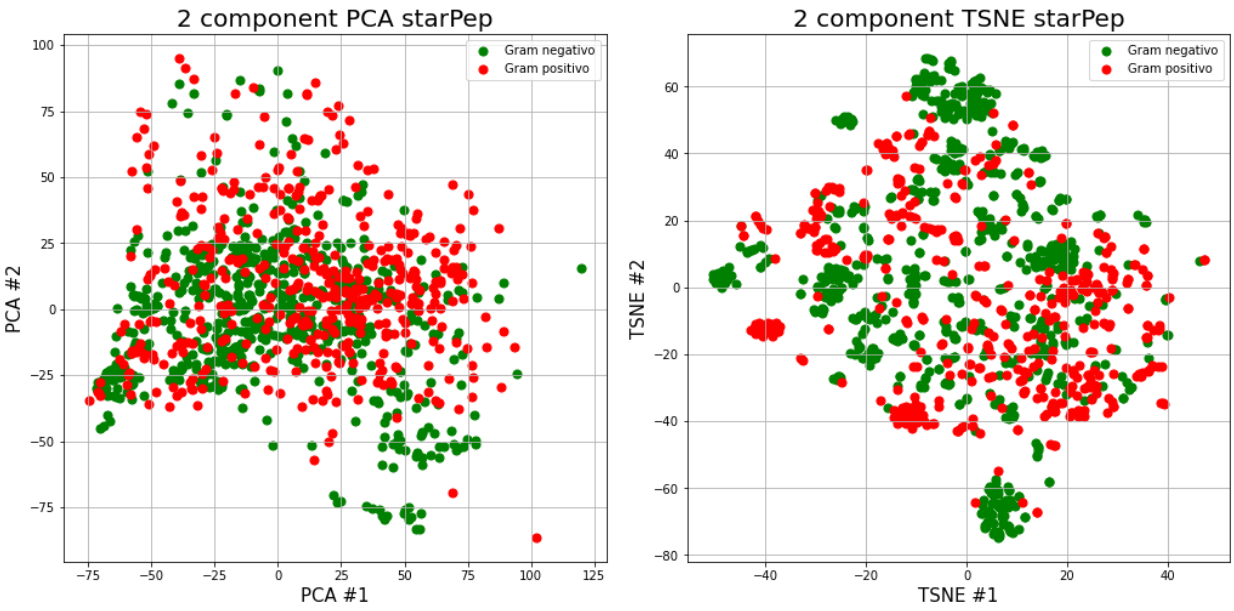
Lo primero que podemos observar en las proyecciones de la Figura 21, es que los

conjuntos no son linealmente separables. Es decir, tanto el conjunto Gram negativo como el conjunto Gram positivo no están limitados a un solo cuadrante, sino que se extienden a lo largo de toda la proyección. Esto puede indicar que no existe una sola característica que predomine en la representación, sino que más bien, la diferenciación de ambos conjuntos depende de múltiples características. También podemos observar que no existe una superposición fuerte entre las muestras proyectadas. A su vez, es claramente observable que en las regiones donde coexisten las muestras Gram positivas con las muestras Gram negativas, siempre predomina en densidad una de las dos. Dando a entender que aunque existen características que pueden compartirse entre ambos conjuntos (por ejemplo, la inclusión del aminoácido alanina en su secuencia), siempre será más predominante en uno de estos dos.

Posteriormente se decidió generar una proyección de los mismos conjuntos que se utilizaron en la Figura 21, pero en este caso utilizando la representación generada por los descriptores moleculares de starPep. Se eligió la representación de starPep, debido a que esta captura la información de las propiedades fisicoquímicas de los péptidos. Esto es importante, debido a que esperamos que los péptidos con diferentes propiedades fisicoquímicas presenten actividades biológicas diferentes. Lo cual debe ser el caso para los conjuntos Gram positivo y Gram negativo. Las proyecciones PCA y TSNE de esta representación se pueden observar en la Figura 22.

En las proyecciones de la Figura 22, podemos observar un comportamiento similar al que se presentó en la Figura 21 con la composición de aminoácidos. No se observa que los conjuntos sean linealmente separables, pero sí se observa que existe una predominancia de estos conjuntos en diferentes regiones del espacio. En este caso, se observa que la superposición entre el conjunto Gram negativo y Gram positivo es mucho menor que en la Figura 21; y en el caso del ejemplo de TSNE, se observa una predominancia más marcada de estos conjuntos en diferentes regiones. Esto puede indicar que existen un determinado conjunto de características fisicoquímicas, donde las muestras Gram negativas y Gram positivas difieren de manera significativa.

Por último, se generó una proyección de estos conjuntos utilizando la representación de la red neuronal BERT-ESM. Se eligió esta representación debido a que Devlin *et al.* (2019) afirman que su representación de las secuencias es capaz de representar las capacidades fisicoquímicas de las secuencias. También, se presenta evidencia



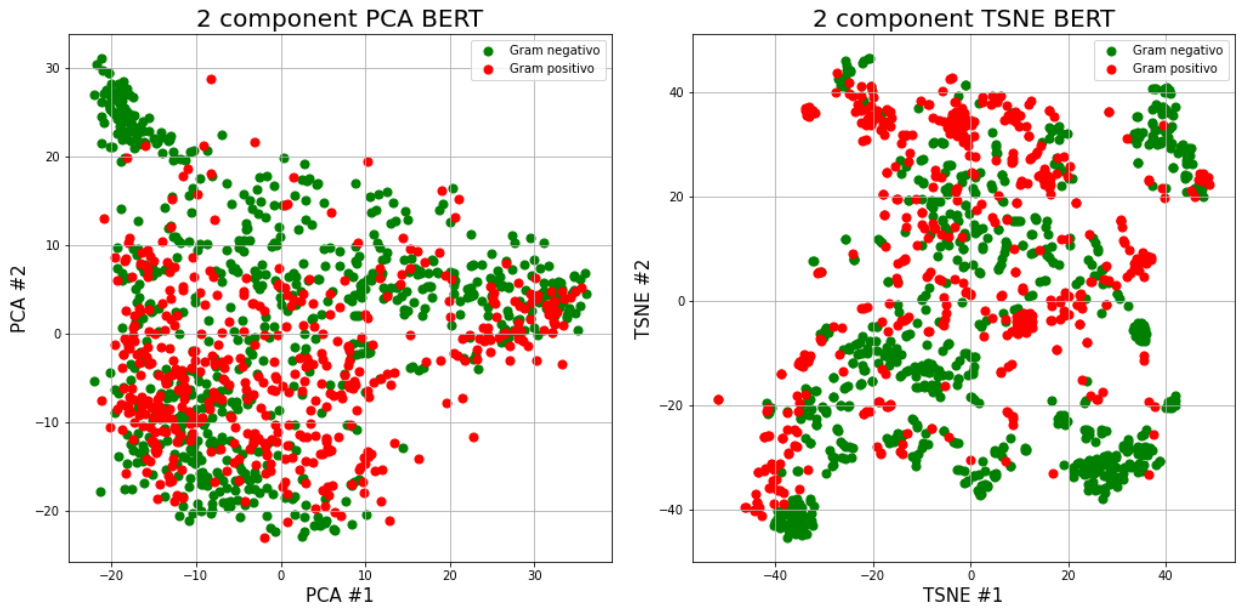
(a) (b)

Figura 22. PCA y TSNE de starPep. (a)Representación de los conjuntos de entrenamiento utilizando descriptores de starPep y PCA. (b) Representación de los conjuntos de entrenamiento utilizando descriptores de starPep y TSNE.

que indican que la distancia en esta representación podría estar correlacionada con similitud basada en alineamiento de secuencias. Las proyecciones PCA y TSNE de esta representación se pueden observar en la Figura 23.

En la proyección de BERT-ESM, podemos observar una discrepancia con la tendencia observada en las proyecciones anteriores. En el caso de la proyección generada por PCA, podemos observar una fuerte superposición de los conjuntos Gram positivo y Gran negativo, aunque con una marcada tendencia por parte de los datos del conjunto Gram positivo a mantenerse más cercano al eje “X” (PCA 1) de la gráfica. Sin embargo, para la proyección generada por TSNE, aunque se observa más dispersión de datos que en las proyecciones anteriores, no se observa una superposición entre el conjunto Gram negativo y Gram positivo. En este caso particular, se cree que al estar codificando más de una sola característica de los péptidos, es que el algoritmo PCA fue más sensible y terminó por representar ambos conjuntos más superpuestos que en los casos anteriores.

Las tres proyecciones anteriores son una fuerte evidencia de que existen diferencias significativas en la composición de aminoácidos y propiedades fisicoquímicas de



(a) (b)

Figura 23. PCA y TSNE de BERT. (a)Representación de los conjuntos de entrenamiento utilizando descriptores de BERT y PCA. (b) Representación de los conjuntos de entrenamiento utilizando descriptores de BERT y TSNE.

las secuencias pertenecientes a los conjuntos Gram positivo y Gram negativos. Sin embargo, con esta experimentación no podemos afirmar que ambos conjuntos son totalmente separables.

4.2. Modelos basados en algoritmos de clasificación de una sola clase

Uno de los objetivos planteados al inicio de esta tesis, fue el de encontrar los mejores modelos de clasificación de una clase, al ser implementados en la clasificación de péptidos antimicrobianos con la metodología que se desarrollara, en este caso es la descrita en el Capítulo 3. Para esto se requirió de una serie de experimentos para conocer el comportamiento de los modelos basados en clasificadores de una clase. Primero, se realizó un experimento para conocer cómo se comportan los modelos de clasificación de una clase. Segundo, se realizaron experimentos para conocer cómo es que estos modelos se comportan utilizando la metodología de clasificación jerárquico. Por último, se realizó una serie de experimentos utilizando descriptores reportados en la literatura.

Tabla 13. Resultados AMP y antibacterial. Se muestra el desempeño de los algoritmos de una sola clase para las etapas AMP y antibacterial, utilizando las métricas de sensibilidad (“target”) y especificidad (“outlier”).

Clasificador	Clase	AMP			Antibacterial		
		Euclidean	Chebyshev	Manhattan	Euclidean	Chebyshev	Manhattan
Bagging ISF	target	80.2%	81.2%	82.4%	83.5%	81.2%	84.5%
	outlier	38.1%	38.3%	38.9%	40.5%	38.3%	40.1%
Bagging LOF	target	99.0%	99.5%	99.4%	99.3%	99.5%	99.1%
	outlier	27.4%	26.4%	26.8%	26.9%	26.4%	27.0%
Bagging BayesNet	target	88.1%	89.0%	89.4%	88.0%	89.0%	90.7%
	outlier	45.1%	44.8%	44.8%	44.5%	44.8%	43.9%

4.2.1. Experimento #1 - Utilización individual de los algoritmos de clasificación de una sola clase

En este experimento se evaluó de manera individual el desempeño de los algoritmos de una sola clase seleccionados para generar los modelos. Este experimento busca contestar una pregunta puntual: ¿Cómo se desempeñan los algoritmos de clasificación de una sola clase, al ser entrenados con los conjuntos de la Tabla 11?. A su vez, con base en los resultados obtenidos en el presente experimento, se busca encontrar un modelo para ser utilizado como el primer filtro, en el paradigma de clasificación jerárquico propuesto en la sección 3.7.

Los modelos que se probaron fueron: “OneClass Bagging BayesNet”, “Bagging Isolation Forest” y “Bagging Local Outlier Factor”. Los modelos individuales se construyeron utilizando los conjuntos de entrenamiento de la Tabla 11 y los descriptores moleculares de “starPep”, utilizando el filtro de outliers de la sección 3.4. Los modelos se validaron utilizando los conjuntos de validación externos de la Tabla 12. Los resultados de este proceso se muestran en las tablas 13 y 14. En las tablas 13 y 14, los resultados están presentados de forma vertical, presentando primero un porcentaje haciendo referencia a una clase denominada “target” y consecuentemente una denominada “outlier”. En este caso, la clase “target” hace referencia a la recuperación de la clase positiva (sensibilidad), y la clase “outlier” hace referencia a la recuperación de la clase negativa (especificidad). También, se presentan tres columnas de resultados para cada modelo, los denominados “Euclidean”, “Chebychev” y “Manhattan”; estos hacen referencia a la distancia que utilizó el filtro de outliers propuesto en la sección 3.4.

De las tablas 13 y 14 podemos hacer el siguiente análisis. Primero, los modelos de

Tabla 14. Resultados Gram negativo y Gram positivo. Se muestra el desempeño de los algoritmos de una sola clase para las etapas Gram positiva y Gram negativa, utilizando las métricas de sensibilidad ("target") y especificidad ("outlier").

Clasificador	Clase	Gram negativo			Gram positivo		
		Euclidean	Chebyshev	Manhattan	Euclidean	Chebyshev	Manhattan
Bagging ISF	target	82.6%	83.3%	84.0%	73.4%	71.3%	74.5%
	outlier	24.0%	21.7%	23.5%	42.2%	41.8%	40.8%
Bagging LOF	target	95.8%	93.8%	93.1%	93.6%	92.6%	94.7%
	outlier	23.7%	26.0%	25.2%	30.1%	33.0%	30.6%
Bagging BayesNet	target	91.7%	88.2%	88.9%	87.2%	91.5%	92.6%
	outlier	41.3%	46.3%	44.4%	39.7%	39.8%	37.3%

clasificación de una clase tienden a tener una sensibilidad relativamente alta y una especificidad relativamente baja, esto es debido a que estos modelos se entrenan utilizando únicamente la información de la clase positiva. Segundo, en la mayoría de los casos, el modelo de "OneClass Bagging BayesNet" tiene la mejor relación entre la sensibilidad y la especificidad. Por esta razón, se decidió que el modelo "OneClass Bagging BayesNet" se utilizaría como el primer filtro en la metodología jerárquica propuesta en la sección 3.7.

4.2.2. Experimento #2 - Evaluación de la clasificación jerárquica

Este experimento se diseñó con la intención de evaluar el desempeño de los algoritmos propuestos utilizando la metodología de clasificación jerárquica propuesta en la sección 3.7. También, se buscó evaluar estos resultados para tomar una decisión, sobre el modelo a utilizar en la segunda etapa de filtrado de la metodología de clasificación jerárquica.

En esta segunda etapa, no se está utilizando un solo modelo, sino que se entrena un ensamble de modelos, donde todos utilizan el mismo algoritmo de clasificación. En este caso, se generó un modelo para cada uno de los "clusters" que se encuentren por el proceso de "Clustering" descrito en la sección 3.6. Por nombrar un ejemplo, supongamos un caso donde se utilizó este proceso de "Clustering" sobre un conjunto "X", encontrando seis "clusters". En este caso, para la segunda etapa se generarían seis modelos, cada uno utilizando la información individual de cada clúster. Los modelos se generan utilizando el mismo algoritmo (Isolation Forest, Local Outlier Factor, BayesNet) para cada uno los subconjuntos.

Tabla 15. Resultados del flujo de clasificación Gram negativo. Se muestra el desempeño de la clasificación de los modelos Bagging Bayes Net, Bagging Outlier Factor y Bagging Isolation Forest para el flujo Gram Negativo, utilizando las métricas de sensibilidad ("target") y especificidad (“outlier”).

Clasificador	Clase	EM				K-means-Euclidean				K-means-Manhattan			
		AMP	Antibac	Gram -	Jerárquico	AMP	Antibac	Gram -	Jerárquico	AMP	Antibac	Gram -	Jerárquico
Bagging ISF	target	64.39%	69.55%	27.45%	21.56%	62.84%	67.49%	22.49%	18.16%	67.59%	66.97%	20.43%	16.82%
	outlier	63.59%	65.72%	70.38%	84.60%	69.55%	71.32%	85.72%	93.92%	67.49%	68.25%	78.82%	90.91%
Bagging LOF 5-5	target	77.91%	80.70%	58.30%	48.09%	79.56%	81.11%	58.30%	49.53%	78.63%	80.08%	50.15%	42.93%
	outlier	52.97%	51.79%	52.09%	68.67%	56.40%	53.39%	53.51%	71.15%	56.87%	54.10%	59.76%	75.57%
Bagging LOF 5-10	target	78.84%	80.90%	61.19%	50.98%	79.15%	80.80%	58.30%	49.22%	79.05%	80.28%	52.42%	44.89%
	outlier	52.44%	51.20%	48.73%	66.43%	55.57%	52.97%	53.51%	70.26%	56.81%	53.80%	57.46%	75.04%
Bagging LOF 5-15	target	78.63%	80.80%	60.37%	50.15%	79.36%	80.80%	56.55%	47.98%	79.25%	79.97%	49.84%	42.93%
	outlier	52.44%	51.15%	48.25%	66.01%	55.75%	52.86%	53.15%	70.14%	56.57%	53.98%	62.18%	78.82%
Bagging BayesNet	target	79.36%	77.60%	62.64%	52.21%	77.08%	78.63%	59.99%	52.35%	77.81%	79.05%	59.13%	51.59%
	outlier	57.52%	55.22%	45.13%	67.02%	58.46%	58.76%	47.98%	71.03%	58.17%	56.46%	57.46%	75.39%

El experimento consistió en seguir el flujo de clasificación jerárquica descrito en la sección 3.7, utilizando como conjuntos de validación los conjuntos de la Tabla 11. Los algoritmos utilizados en este experimento fueron: “Bagging Isolation Forest”, “Bagging BayesNet” y “Bagging Local Outlier Factor” utilizando como “número de vecinos” los intervalos de 5 a 5, 5 a 10 y 5 a 15. En este caso particular, nos interesó variar el número de vecinos utilizados por el algoritmo “Local Outlier Factor”. Esto para evaluar si esta variación de parámetros tiene o no un impacto significativo en el desempeño del modelo.

Los resultados de este experimento se encuentran en las tablas de la 15 a la 18. De forma similar a las tablas presentadas en el Experimento 1, en las tablas de la 15 a la 18, los renglones denominados “Target” hacen referencia a la sensibilidad mientras que los renglones denominados “Outlier” hacen referencia a la especificidad. En este caso, se partieron los resultados en tres conjuntos, los que se obtuvieron utilizando el algoritmo de “clustering” “Expectation maximization” (EM), y los que se obtuvieron utilizando el algoritmo “K-means”, con distancias “Euclidean” y “Manhattan" (ver sección 3.6). Para cada uno de estos conjuntos se presenta los resultados de las etapas AMP, antibacterial, Gram positivo y Gram negativo de manera individual, así como una última columna donde se presenta el resultado total del flujo jerárquico.

En general, el resultado de esta experimentación es la obtención de modelos con una especificidad más alta que la sensibilidad, en contraste con los resultados del Experimento #1 donde la sensibilidad fue siempre más alta que la especificidad. Este aumento en la especificidad proviene de la segmentación del conjunto de entrenamiento en subconjuntos generados por el proceso de “Clustering”. En este caso, al generar más modelos, entrenados con menos secuencias, estamos generando un mo-

Tabla 16. Resultados del flujo de clasificación Gram negativo validando con conjuntos externos. Se muestra el desempeño de la clasificación de los modelos Bagging Bayes Net, Bagging Outlier Factor y Bagging Isolation Forest para el flujo Gram Negativo Externo, utilizando las métricas de sensibilidad ("target") y especificidad (“outlier”).

Clasificador	Clase	EM				K-means-Euclidean				K-means-Manhattan			
		AMP	Antibac	Gram -	Jerárquico	AMP	Antibac	Gram -	Jerárquico	AMP	Antibac	Gram -	Jerárquico
Bagging ISF	target	70.33%	67.29%	31.42%	21.71%	71.80%	69.58%	24.12%	19.92%	73.21%	68.92%	21.08%	16.41%
	outlier	57.74%	64.80%	53.33%	80.93%	66.06%	67.04%	72.40%	90.17%	63.56%	66.33%	62.81%	87.95%
Bagging LOF 5-5	target	86.05%	82.83%	60.71%	52.22%	86.56%	82.95%	62.24%	55.35%	86.45%	82.36%	56.64%	49.49%
	outlier	47.55%	50.56%	34.63%	64.02%	50.47%	49.02%	35.60%	64.65%	50.25%	50.14%	43.25%	67.70%
Bagging LOF 5-10	target	86.24%	83.52%	64.12%	55.35%	86.15%	82.67%	63.40%	56.01%	85.90%	82.55%	59.83%	52.03%
	outlier	47.31%	50.32%	30.85%	62.16%	50.29%	49.20%	34.43%	64.15%	50.32%	49.89%	40.18%	66.40%
Bagging LOF 5-15	target	86.02%	83.27%	64.22%	55.38%	86.09%	82.64%	62.31%	55.04%	85.90%	82.39%	58.86%	50.65%
	outlier	47.09%	50.28%	30.64%	61.87%	50.26%	49.15%	34.50%	64.13%	50.02%	49.85%	44.17%	69.70%
Bagging BayesNet	target	82.11%	80.67%	69.86%	57.26%	81.61%	82.39%	61.21%	53.85%	81.89%	82.20%	67.85%	58.77%
	outlier	52.77%	53.10%	28.11%	63.24%	55.67%	56.18%	48.20%	69.85%	54.82%	54.87%	35.33%	69.58%

Tabla 17. Resultados del flujo de clasificación Gram positivo. Se muestra el desempeño de la clasificación de los modelos Bagging Bayes Net, Bagging Outlier Factor y Bagging Isolation Forest para el flujo Gram Positivo, utilizando las métricas de sensibilidad ("target") y especificidad (“outlier”).

Clasificador	Clase	EM				K-means-Euclidean				K-means-Manhattan			
		AMP	Antibac	Gram +	Jerárquico	AMP	Antibac	Gram +	Jerárquico	AMP	Antibac	Gram +	Jerárquico
Bagging ISF	target	64.39%	69.55%	19.29%	14.55%	62.84%	67.49%	31.57%	25.59%	67.59%	66.97%	29.61%	24.14%
	outlier	63.59%	65.72%	92.09%	94.86%	69.55%	71.32%	85.89%	92.44%	67.49%	68.25%	87.43%	92.68%
Bagging LOF 5-5	target	77.91%	80.70%	60.37%	49.22%	79.56%	81.11%	57.27%	48.91%	78.63%	80.08%	58.10%	46.85%
	outlier	52.97%	51.79%	67.72%	77.87%	56.40%	53.39%	62.71%	75.22%	56.87%	54.10%	71.15%	80.88%
Bagging LOF 5-10	target	78.84%	80.90%	62.12%	50.87%	79.15%	80.80%	57.99%	48.81%	79.05%	80.28%	57.68%	47.16%
	outlier	52.44%	51.20%	65.36%	75.87%	55.57%	52.97%	60.88%	73.74%	56.81%	53.80%	69.38%	79.64%
Bagging LOF 5-15	target	78.63%	80.80%	60.37%	50.25%	79.36%	80.80%	58.20%	49.01%	79.25%	79.97%	57.68%	47.67%
	outlier	52.44%	51.15%	65.30%	75.75%	55.75%	52.86%	60.11%	73.45%	56.57%	53.98%	68.79%	79.05%
Bagging BayesNet	target	79.36%	77.60%	65.32%	54.28%	77.08%	78.63%	68.42%	57.68%	77.81%	79.05%	70.48%	57.99%
	outlier	57.52%	55.22%	57.58%	70.50%	58.46%	58.76%	49.43%	69.32%	58.17%	56.46%	56.28%	72.21%

Tabla 18. Resultados del flujo de clasificación Gram positivo validando con conjuntos externos. Desempeño de la clasificación de los modelos Bagging Bayes Net, Bagging Outlier Factor y Bagging Isolation Forest para el flujo Gram Positivo Externo, utilizando las métricas de sensibilidad ("target") y especificidad (“outlier”).

Clasificador	Clase	EM				K-means-Euclidean				K-means-Manhattan			
		AMP	Antibac	Gram +	Jerárquico	AMP	Antibac	Gram +	Jerárquico	AMP	Antibac	Gram +	Jerárquico
Bagging ISF	target	70.33%	67.29%	23.00%	18.64%	71.80%	69.58%	41.44%	34.14%	73.21%	68.92%	36.52%	31.01%
	outlier	57.74%	64.80%	85.00%	91.29%	66.06%	67.04%	79.20%	89.12%	63.56%	66.33%	77.90%	89.55%
Bagging LOF 5-5	target	86.05%	82.83%	66.38%	54.60%	86.56%	82.95%	69.98%	59.11%	86.45%	82.36%	65.91%	56.04%
	outlier	47.55%	50.56%	63.02%	74.06%	50.47%	49.02%	54.32%	68.44%	50.25%	50.14%	60.86%	73.07%
Bagging LOF 5-10	target	86.24%	83.52%	67.54%	55.92%	86.15%	82.67%	70.77%	59.80%	85.90%	82.55%	67.32%	57.48%
	outlier	47.31%	50.32%	59.19%	71.65%	50.29%	49.20%	51.96%	67.12%	50.32%	49.89%	57.60%	70.87%
Bagging LOF 5-15	target	86.02%	83.27%	67.16%	55.88%	86.09%	82.64%	70.92%	59.99%	85.90%	82.39%	67.79%	58.05%
	outlier	47.09%	50.28%	59.05%	71.43%	50.26%	49.15%	51.38%	66.66%	50.02%	49.85%	56.94%	70.52%
Bagging BayesNet	target	82.11%	80.67%	77.47%	62.50%	81.61%	82.39%	81.10%	67.07%	81.89%	82.20%	81.42%	67.01%
	outlier	52.77%	53.10%	52.51%	67.44%	55.67%	56.18%	39.80%	66.20%	54.82%	54.87%	47.47%	68.85%

delo con un espacio de conocimiento más específico. Después de evaluar los modelos, se decidió utilizar el modelo construido con “Bagging Isolation Forest” dado que este es el que obtiene la especificidad más alta en todos los casos (entre 80 % y 90 %).

La razón para decidir utilizar estos modelos es debido a que se considera que los modelos con una especificidad alta tienen una menor tendencia a producir falsos positivos; es decir, aunque estos modelos recuperan una cantidad menor de muestras, tenemos una mayor confianza en que las muestras recuperadas son antimicrobianas y no falsos positivos. Se considera que esto podría ser de utilidad en caso real de cribado virtual de secuencias. Esto porque en aplicaciones reales, es común que se analice un conjunto grande de secuencias; y que a su vez el objetivo sea reducir el número de candidatos lo más posible.

4.2.3. Experimento #3 - utilización de descriptores de la literatura

Una vez definidos los algoritmos a utilizar en las etapas de clasificación (Bagging BayesNet y Bagging Isolation Forest), se planteó realizar un experimento donde se utilizarán descriptores moleculares de la literatura, además de los de “starPep” que fueron previamente utilizados. Los descriptores moleculares utilizados fueron: starPep (Aguilera-Mendoza *et al.* (2019)), ProtDcal (Romero-Molina *et al.* (2019)), iFeature (Chen *et al.* (2018)), BERT ESM (Rives *et al.* (2019)). Se generaron en total 8 modelos a partir de los 4 conjuntos de descriptores mencionados. A continuación se describen estos modelos y se enumeran conforme a su aparición en la Tabla 19.

El primer modelo (starPep #1) está basado en descriptores moleculares de starPep, cuyos descriptores se filtraron utilizando el proceso de selección de características descrito en la sección 3.4, utilizando un filtro de correlación de Spearman al 90 %. El segundo modelo (starPep #2), está construido de manera similar al primero, utilizando el filtro de correlación de Spearman al 95 %. El Tercer modelo (iFeature), se generó utilizando los descriptores moleculares del mismo nombre, utilizando el proceso de selección de características descrito en la sección 3.4, utilizando a su vez, un filtro de correlación de Spearman al 90 %.

El cuarto y quinto modelo (starPep+iFeature #1 y starPep+iFeature #2, respecti-

Tabla 19. Modelos generados para la experimentación con descriptores moleculares. Se enlistan todos los modelos generados durante la experimentación, así como el número de descriptores utilizados por etapa y si estos modelos se filtraron o no.

Modelo	Gram -/+	Descriptores por etapa			Filtro de descriptores?
		AMP	AB	Gram	
starPep #1	Gram -	93	97	52	Spearman 90 %
	Gram +			85	
starPep #2	Gram -	111	104	52	Spearman 95 %
	Gram +			79	
iFeature	Gram -	35	34	27	Spearman 90 %
	Gram +			33	
Starpep + iFeature #1	Gram -	128	131	79	No se utilizó
	Gram +			118	
Starpep + iFeature #2	Gram -	30	27	21	Spearman 90 %
	Gram +			37	
ProtDcal	Gram -	207	93	93	No se utilizó
	Gram +			93	
BERT-ESM #1	Gram -	7	6	6	Sperman 90 %
	Gram +			7	
BERT-ESM #2	Gram -	1280	1280	1280	No se utilizó
	Gram +			1280	

vamente), se generaron utilizando una combinación de los descriptores moleculares de starPep e iFeature. El cuarto modelo (starPep+iFeature #1), se generó utilizando una combinación de los descriptores seleccionados por el proceso de selección de características, para el primer (starPep #1) y tercer (iFeature) modelo. El quinto modelo (starPep+iFeature #2), se generó utilizando una combinación de todos los descriptores moleculares de starPep e iFeature, a la que posteriormente se le aplicó el proceso de selección de características de la sección 3.4, utilizando un filtro de Spearman al 90 %.

El sexto modelo (protDcal) se generó a partir de los descriptores del mismo nombre, sin utilizar un proceso de selección de características, debido a que este ya habia sido previamente implementado por Pinacho-Castellanos *et al.* (2021a) (ver sección 3.2.2). Por último, se construyeron dos modelos, séptimo y octavo, basados en la representación de la red BERT-ESM. El séptimo (BERT-ESM #1) se construyó utilizando un filtro de selección de características con un valor de Spearman al 90 %. El octavo modelo (BERT-ESM #2) utilizó la representación completa generada por BERT-ESM (1280 características). A partir de estos ocho modelos, es que se generaron tanto un flujo jerárquico de detección de muestras Gram negativas como Gram positivas.

En las tablas 20 y 21 se presentan los resultados obtenidos para el flujo de detec-

Tabla 20. Mejores resultados para el flujo de detección de muestras Gram negativas. Se muestran únicamente los resultados de aquellos modelos que tuviesen el resultado más alto de coeficiente de correlación de Matthews. También se indica el tipo de filtro y de cluster que se utilizó para generar el modelo.

Tipo de distancia / Clúster:		Etapas				Distancia utilizada en LOF	Tipo de clúster utilizado
		AMP	AB	Gram-	Jer		
starPep #1	Target	75.69	70.83	29.86	25.0	Chebyshev	K-means Euclidean
	Outlier	61.17	73.68	95.16	97.34		
	MCC	0.20	0.26	0.26	0.29		
starPep #2	Target	75.0	72.22	27.08	22.22	Chebyshev	K-means Manhattan
	Outlier	63.53	75.98	99.76	99.82		
	MCC	0.21	0.28	0.47	0.43		
iFeature	Target	77.62	79.72	54.54	51.04	Euclidean	K-means Manhattan
	Outlier	13.16	11.62	25.50	29.33		
	MCC	-0.07	-0.07	-0.12	-0.11		
starPep + iFeature #1	Target	74.30	78.47	40.27	37.5	Euclidean	K-means Euclidean
	Outlier	52.21	62.59	75.63	89.55		
	MCC	0.14	0.22	0.09	0.21		
starPep + iFeature #2	Target	76.38	73.61	42.36	33.33	Chebyshev	K-means Euclidean
	Outlier	46.66	49.55	80.82	90.02		
	MCC	0.12	0.12	0.15	0.19		
ProtDcal	Target	93.05	81.94	59.72	55.55	Chebyshev	K-means Euclidean
	Outlier	57.94	78.70	93.56	93.74		
	MCC	0.27	0.37	0.46	0.43		
BERT-ESM #1	Target	70.83	70.83	39.58	20.13	Euclidean	K-means Manhattan
	Outlier	26.43	39.76	81.47	85.60		
	MCC	-0.01	0.05	0.14	0.04		
BERT-ESM #2	Target	63.88	72.22	8.33	7.63	Euclidean	EM
	Outlier	43.89	44.24	99.70	99.70		
	MCC	0.04	0.08	0.22	0.21		

ción de muestras Gram negativas, utilizando para la validación, los conjuntos: Gram negativo, Positivo externo, Negativo y Negativo externo. En estas tablas se presentan únicamente los mejores modelos generados para cada conjunto de descriptores, donde: “target” hace referencia a la sensibilidad, “outlier” hace referencia a especificidad y “MCC” al coeficiente de correlación de Matthews. También se presenta para cada modelo, el tipo de distancia que se utilizó en el filtrado de outliers, y el tipo de “cluster” del proceso de “clustering”. Los resultados completos de la experimentación se presentan en el Anexo B.

En general, podemos observar que la mayoría de los modelos presentan la misma tendencia; una especificidad que supera el 90 % y una sensibilidad que oscila entre el 20 % y el 50 %. Este comportamiento se debe principalmente a dos razones; primero, al utilizar tres modelos de detección de anomalías en serie, se aumenta la probabilidad de descartar una muestra. Segundo, contamos con una menor representación (tamaño de muestra) para las etapas Gram negativo y Gram positivo, lo cual hace menos probable la recuperación en estas etapas. Los modelos generados con starPep, están ambos en un intervalo muy cercano en cuanto a la recuperación que presentaron, una sensibilidad cercana al 25 % y una especificidad muy por encima del 90 %. Lo anterior,

Tabla 21. Mejores resultados para el flujo de detección de muestras Gram negativas utilizando los conjuntos externos de validación. Se muestran únicamente los resultados de aquellos modelos que tuviesen el resultado más alto de coeficiente de correlación de Matthews. También se indica el tipo de filtro y de cluster que se utilizó para generar el modelo.

Tipo de distancia / Clúster:		Etapas				Distancia utilizada en LOF	Tipo de clúster utilizado
		AMP	AB	Gram-	Jer		
starPep #1	Target	73.56	74.65	23.99	23.01	Manhattan	K-means Euclidean
	Outlier	62.44	64.70	84.91	92.06		
	MCC	0.33	0.36	0.10	0.21		
starPep #2	Target	74.94	53.97	12.17	9.11	Euclidean	K-means Manhattan
	Outlier	66.32	84.17	91.02	97.56		
	MCC	0.38	0.39	0.05	0.14		
iFeature	Target	76.51	76.12	41.82	32.37	Manhattan	K-means Manhattan
	Outlier	5.50	5.54	18.25	23.34		
	MCC	-0.26	-0.26	-0.40	-0.43		
starPep + iFeature #1	Target	75.58	66.83	26.41	23.75	Euclidean	K-means Manhattan
	Outlier	44.16	55.17	61.65	79.55		
	MCC	0.19	0.21	-0.12	0.03		
starPep + iFeature #2	Target	80.31	72.71	23.32	20.54	Chebyshev	K-means Euclidean
	Outlier	34.57	34.16	76.02	85.10		
	MCC	0.15	0.07	-0.00	0.07		
ProtDcal	Target	91.13	79.43	45.76	45.30	Euclidean	K-means Euclidean
	Outlier	48.57	67.73	83.23	84.25		
	MCC	0.39	0.45	0.31	0.32		
BERT-ESM #1	Target	66.46	54.32	27.23	15.02	Manhattan	EM
	Outlier	39.60	56.95	81.93	90.68		
	MCC	0.05	0.10	0.10	0.08		
BERT-ESM #2	Target	64.80	61.15	3.60	2.61	Euclidean	K-means Manhattan
	Outlier	40.87	36.70	99.85	99.92		
	MCC	0.05	-0.02	0.14	0.12		

puede indicar que estos valores son una cota superior para la utilización de este conjunto de descriptores al ser utilizados con el proceso de selección de características de la sección 3.4.

El modelo generado con iFeature, es el único de los modelos que presenta resultados muy distintos a la media de los valores obtenidos por los demás modelos. Los resultados obtenidos son muy similares a los de ProtDcal en sensibilidad, con la desventaja de tener una especificidad muy por debajo de lo que presentan el resto de modelos. En el caso donde se analizan los modelos que se generaron utilizando una combinación de descriptores moleculares de starPep e iFeature, notamos que los modelos aumentan su sensibilidad en 10 % a partir de una pérdida similar en la especificidad.

El mejor modelo de todos los generados durante todo el proceso de experimentación, fue el generado con descriptores de ProtDcal, esto debido a que presenta el menor compromiso entre la sensibilidad y la especificidad dado que en las tres etapas del modelo (AMP, antibacteriano y Gram negativo) mantiene el porcentaje más alto de sensibilidad. Por último, los modelos generados utilizando la representación de BERT-ESM obtuvieron valores similares a starPep, con una especificidad muy alta,

Tabla 22. Mejores resultados para el flujo de detección de muestras Gram positivas. Se muestran únicamente los resultados de aquellos modelos que tuviesen el resultado más alto de coeficiente de correlación de Matthews. También se indica el tipo de filtro y de cluster que se utilizó para generar el modelo.

Tipo de distancia / Clúster:		Etapas				Distancia utilizada en LOF	Tipo de clúster utilizado
		AMP	AB	Gram-	Jer		
starPep #1	Target	67.02	65.95	26.59	26.59	Manhattan	EM
	Outlier	60.47	62.71	96.63	97.22		
	MCC	0.12	0.13	0.24	0.27		
starPep #2	Target	59.57	48.93	20.21	19.14	Chebyshev	K-means Manhattan
	Outlier	64.60	80.53	99.17	99.52		
	MCC	0.11	0.16	0.32	0.34		
iFeature	Target	69.89	70.96	43.01	39.78	Manhattan	K-means Euclidean
	Outlier	15.64	16.41	38.25	40.79		
	MCC	-0.08	-0.07	-0.08	-0.08		
starPep + iFeature #1	Target	71.27	63.82	28.72	28.72	Euclidean	K-means Manhattan
	Outlier	48.49	60.0	92.62	93.80		
	MCC	0.08	0.10	0.17	0.19		
starPep + iFeature #2	Target	73.40	77.65	29.78	25.53	Euclidean	K-means Euclidean
	Outlier	46.84	49.91	96.93	97.22		
	MCC	0.09	0.12	0.28	0.26		
ProtDcal	Target	87.23	74.46	26.59	25.53	Manhattan	K-means Manhattan
	Outlier	57.40	77.40	98.87	98.99		
	MCC	0.20	0.26	0.36	0.36		
BERT-ESM #1	Target	65.95	65.95	39.36	30.85	Euclidean	K-means Euclidean
	Outlier	33.21	48.61	70.67	81.23		
	MCC	-0.00	0.06	0.04	0.06		
BERT-ESM #2	Target	63.82	59.57	20.21	12.76	Manhattan	K-means Manhattan
	Outlier	37.40	35.33	97.52	97.58		
	MCC	0.00	-0.02	0.21	0.13		

aunque estos modelos no superan a los generados con starPep. La baja sensibilidad de estos modelos se debe específicamente a que tanto en el modelo BERT-ESM #1 como en el modelo BERT-ESM #2, se tienen valores bajos de especificidad para la etapa de detección de péptidos Gram negativos.

En las tablas 22 y 23 se presentan los resultados obtenidos para el flujo de detección de muestras Gram positivas, utilizando para la validación, los conjuntos: Gram negativo, Positivo externo, Negativo y Negativo externo. En estas tablas se presentan únicamente los mejores modelos generados para cada conjunto de descriptores, donde: “target” hace referencia a la sensibilidad, “outlier” hace referencia a la especificidad y “MCC” al coeficiente de correlación de Matthews. También se presenta para cada modelo, el tipo de distancia que se utilizó en el filtrado de outliers, y el tipo de “cluster” del proceso de “clustering”. Los resultados completos de la experimentación se presentan en el Anexo B.

Los modelos construidos para la detección de muestras Gram positivas presentan un comportamiento muy similar entre sí, a excepción del modelo generado con descriptores moleculares de iFeature, el cual obtiene una recuperación superior a estos por un 10 %, sin embargo, con una especificidad mucho menor. En general, los mo-

Tabla 23. Mejores resultados para el flujo de detección de muestras Gram positivas utilizando los conjuntos externos de validación. Se muestran únicamente los resultados de aquellos modelos que tuviesen el resultado más alto de coeficiente de correlación de Matthews. También se indica el tipo de filtro y de cluster que se utilizó para generar el modelo.

Tipo de distancia / Clúster:		Etapas				Distancia utilizada en LOF	Tipo de clúster utilizado
		AMP	AB	Gram-	Jer		
starPep #1	Target	67.64	58.84	32.44	24.02	Euclidean	K means-Euclidean
	Outlier	69.54	75.84	81.33	91.66		
	MCC	0.35	0.34	0.15	0.21		
starPep #2	Target	75.52	63.79	16.17	14.10	Chebyshev	K means-Manhattan
	Outlier	66.08	76.83	96.54	97.85		
	MCC	0.39	0.39	0.22	0.23		
iFeature	Target	76.79	68.40	46.20	31.35	Chebyshev	EM
	Outlier	4.52	10.97	22.54	31.52		
	MCC	-0.28	-0.25	-0.31	-0.35		
starPep + iFeature #1	Target	74.96	69.99	38.16	35.31	Chebyshev	K means-Euclidean
	Outlier	41.02	55.96	71.34	78.29		
	MCC	0.14	0.21	0.10	0.14		
starPep + iFeature #2	Target	0.14	0.05	0.04	0.07	Manhattan	K means-Manhattan
	Outlier	81.48	73.92	21.12	19.72		
	MCC	32.77	30.00	89.98	91.88		
ProtDcal	Target	90.40	80.84	25.87	25.70	Manhattan	K means-Manhattan
	Outlier	48.37	68.22	98.99	99.01		
	MCC	0.38	0.46	0.40	0.39		
BERT-ESM #1	Target	67.94	52.25	29.64	18.59	Chebyshev	EM
	Outlier	37.99	58.75	74.83	86.16		
	MCC	0.05	0.10	0.04	0.06		
BERT-ESM #2	Target	73.75	62.97	5.58	5.22	Manhattan	K means-Manhattan
	Outlier	41.18	39.42	97.40	97.47		
	MCC	0.14	0.02	0.07	0.07		

delos de clasificación de muestras Gram positivas obtienen un valor de sensibilidad cercano al 25 % y un valor de especificidad superior al 90 %; se intuye que el hecho de que la mayoría de los modelos se acerquen al mismo porcentaje de recuperación es debido principalmente al menor tamaño de representación que se obtuvo para el conjunto de entrenamiento Gram positivo (375 secuencias). El mejor modelo que se obtuvo para este flujo de clasificación fue el generado con descriptores de ProtDcal, aunque a diferencia de los resultados este no supera la sensibilidad de los demás modelos por un margen significativo.

4.3. Comparación con modelos reportados en la literatura

Hasta el momento de escribir este documento, no se ha encontrado un modelo en la literatura que utilice algoritmos de clasificación de una sola clase para clasificar péptidos antimicrobianos. Los modelos que se encuentran en la literatura que clasifican péptidos antimicrobianos utilizan clasificadores de dos o más clases, los cuales requieren de ejemplos de la clase positiva (AMP) y la clase negativa (No-AMP) para entrenarse. Al utilizar algoritmos de clasificación de una sola clase, se utiliza única-

Tabla 24. Resultados de la comparación del modelo AMP-ProtDcal-KME-EUC con los modelos de la literatura. Se muestra los resultados de la predicción del modelo desarrollado y el de los modelos de la literatura, al predecir los conjuntos externos (negativo y positivo).

Modelo	Sensibilidad	Especificidad
AMP – ProtDcal -KME-EUC	91.13	48.97
AMP Scanner (Veltri *et al.* (2018))	90.36	90.79
AMP Discover RNN (Pinacho-Castellanos *et al.* (2021b))	92.31	90.56
AMP Discover RF (Pinacho-Castellanos *et al.* (2021b))	94.77	90.92
amPEP30 (J *et al.* (2020))	34.26	84.20
amPEP30 RF (J *et al.* (2020))	90.47	43.01

mente ejemplos de la clase positiva (AMP) para entrenar modelos; por lo tanto, estos modelos tienden a tener una sensibilidad más alta que la especificidad, a diferencia de los modelos multiclase que tienden a balancear sensibilidad y especificidad.

A pesar de esta diferencia entre ambos tipos de modelos (una clase y binaria), se realizó una comparativa entre diferentes modelos presentados en la literatura y el mejor modelo “AMP” de los presentados. El modelo completo no pudo ser utilizado en la comparación debido a que no se encontró ejemplos en la literatura que clasificaran muestras Gram positivas y Gram negativas. A continuación, se presenta la comparación del modelo aquí propuesto con los modelos presentados en la Tabla 24. Para realizar la comparación, se utilizaron los conjuntos externo positivo y externo negativo que se presentan en la Tabla 12. Estos conjuntos se evaluaron en los servidores web o aplicación, que facilitan cada uno de los modelos con los que se realizó la comparación. Estos resultados se muestran en la Tabla 24.

Como se puede observar en la Tabla 24, el modelo presentado en este documento es equiparable a todos los modelos del estado del arte en sensibilidad, siendo el segundo mejor de la lista solamente superado por el modelo de Pinacho-Castellanos *et al.* (2021a); sin embargo, donde no es equiparable a los modelos del estado del arte es en la métrica de especificidad, aquí, solo supera al modelo basado en “Random forest” presentado por J *et al.* (2020). Incluso si el resultado de especificidad del modelo pueda hacer parecer que el modelo se desempeña peor que los modelos del estado del arte, esto puede ser no necesariamente correcto. Esto se debe a que el conjunto que se utilizó como conjunto negativo en esta prueba no fue validado experimentalmente, sino que se obtuvo como se describe en el Capítulo 3.

Al utilizar un conjunto negativo, no validado para el entrenamiento y prueba de

Tabla 25. Resultados de la utilización de los modelos de una clase en combinación con los modelos binarios de la literatura. Se presenta el resultado de los modelos de una clase, el resultado del modelo binario de la literatura y el resultado de su uso en conjunto. Los resultados se obtienen utilizando los conjuntos externos de validación (positivo y negativo).

Modelo		One Class	Binario (literatura)	One class + Binario
One class + AMP Scanner (Veltri *et al.* (2018))	Target	80.85	90.36	71.06
	Outlier	68.22	90.79	97.29
One class + AB Discover RNN (Pinacho-Castellanos *et al.* (2021b))	Target	80.85	92.14	72.48
	Outlier	68.22	92.09	97.75
One class + AB Discover RF (Pinacho-Castellanos *et al.* (2021b))	Target	80.85	93.96	73.84
	Outlier	68.22	92.33	97.78
One class + amPEP30 (J *et al.* (2020))	Target	80.91	43.47	34.26
	Outlier	68.22	43.01	84.20
One class + amPEP30 RF (J *et al.* (2020))	Target	80.91	90.47	71.29
	Outlier	68.22	43.01	84.20

modelos, existe la posibilidad de que estos contengan moléculas con actividad antimicrobiana. Por lo tanto, el resultado obtenido por el modelo presentado podría ser más preciso que aquellos obtenidos en la literatura. Al utilizarse conjuntos negativos no validados experimentalmente, la especificidad se convierte en un valor de referencia, más que en una medida confiable del comportamiento del modelo. Para demostrar la teoría que sugiere que el conjunto de prueba negativo contiene muestras negativas, se tendría que someter a las secuencias del conjunto a un proceso de validación “In vitro” para determinar si poseen o no propiedades antimicrobianas.

4.4. Utilización de los modelos de una clase en conjunto con clasificadores binarios

Se planteó un experimento donde se utilizó en conjunto, el mejor modelo obtenido de la sección 4.2.3 (ProtDcal) y los modelos de clasificación binaria con los cuales se compararon en la sección anterior. El experimento consistió en utilizar el modelo de una clase construido con descriptores ProtDcal, de manera jerárquica, únicamente hasta la etapa antibacterial (AMP + antibacterial). Una vez se realiza esta primera predicción, se le da a clasificar al modelo de la literatura todas las muestras que el modelo de una clase recupere como positivas. Para este experimento se utilizaron los conjuntos externos positivo y negativo, dado que son los que contienen la mayor cantidad de muestras. Los resultados se presentan en la Tabla 25.

Se obtuvieron resultados muy similares para todos los ejemplos donde se utilizó

el clasificador de una sola clase en conjunto con un modelo binario de la literatura. En general, los modelos pierden alrededor de un 20 % de sensibilidad y ganan desde un 7 % a un 40 % de especificidad, dependiendo del valor inicial de este criterio. Los resultados sugieren que el modelo binario recupera en general más del 90 % de las muestras positivas y que el clasificador de una clase no introduce un sesgo en la recuperación de dichas muestras.

Capítulo 5. Discusiones y conclusión

En el presente capítulo se discuten los resultados obtenidos en función de la metodología utilizada, de forma que se identifiquen los factores clave que afectan al desempeño de los modelos. De la misma manera, se exponen las conclusiones que se obtuvieron a partir de la experimentación, así como también se plantean algunas perspectivas de investigación.

5.1. Discusiones

5.1.1. Conjunto de datos

Para generar los conjuntos de entrenamiento y validación positivos, se planteó una metodología donde se recuperaron secuencias de la base de datos de starPep, las cuales indicaran en sus metadatos una sola actividad. El resultado es la obtención de cuatro conjuntos de secuencias con actividades diferentes: antimicrobiana, antibacterial, Gram positivo y Gram negativo. Como se mencionó en el Capítulo 2, estos conjuntos se generaron para poder usarse en una modalidad de clasificación jerárquica, siendo la actividad antibacteriana, una sub-actividad de la actividad antimicrobiana y a su vez las actividades Gram positiva y Gram negativa, sub-actividades de la actividad antibacterial. En la Figura 24 se presenta un diagrama de esta relación jerárquica.

Al ser utilizado un método de detección de outliers en modalidad jerárquica, surgió la necesidad de generar conjuntos de entrenamiento, cuyas secuencias fueran únicamente representativas de la actividad del conjunto al que pertenecen. Para esto, se generaron los conjuntos antibacterial, Gram positivo y Gram negativo, utilizando secuencias que tienen asociadas únicamente estas actividades y ninguna otra. El único conjunto que contiene secuencias con múltiples actividades es el conjunto AMP, dado que por definición, este conjunto puede contener cualquier tipo de actividad antimicrobiana y por lo tanto cualquier muestra con esta propiedad es representativa de este conjunto.

La razón por la cual se decidió utilizar secuencias que tiene asociadas una única actividad para los conjuntos antibacterial, Gram positivo y Gram negativo, es debido

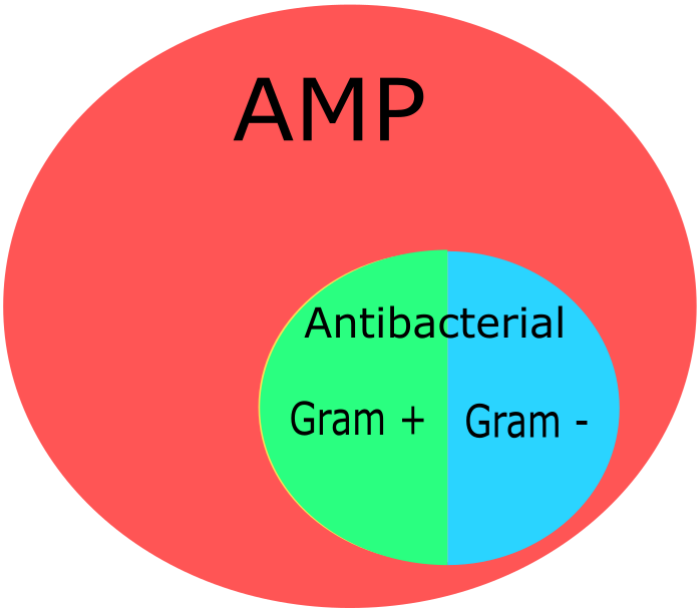


Figura 24. Jerarquía de las actividades biológicas. La etiqueta AMP contiene a cualquier secuencia que posea alguna propiedad antimicrobiana, mientras que la actividad antibacterial contiene a todas las secuencias con actividad anti Gram negativa y anti Gram positiva.

a que si se permitieran secuencias con más de una actividad biológica (antibacterial y antiparasitario por ejemplo), existe la posibilidad de que los algoritmos de detección de anomalías recuperaran secuencias que no poseen la actividad del conjunto que se está utilizando, sino que posean otra actividad.

Por último, en la sección 4.1.2 se propone un experimento para visualizar la representación de los conjuntos Gram negativo y Gram positivo por medio de los algoritmos PCA y TSNE. El objetivo de este experimento fue el de conocer como se distribuían en un espacio de dos dimensiones, los conjuntos de entrenamiento Gram positivo y Gram negativo, una vez que se representaban por medio de descriptores moleculares. En este caso pudimos constatar que los conjuntos Gram positivo y Gram negativo no son linealmente separables. Esto nos indica que ambos conjuntos comparten ciertas características en sus propiedades fisicoquímicas y composición de aminoácidos. Sin embargo, también se observó que aunque no son separables, siempre predomina uno de los dos conjuntos en las diferentes regiones de la proyección. Esto último nos indica que aunque los conjuntos Gram negativo y Gram positivo compartan ciertas características en común, existen otras características en las cuales uno de los dos será más predominante.

5.1.2. Desempeño del modelo de clasificación jerárquica

Para estudiar el desempeño de los clasificadores de una clase, se planteó un experimento donde se utilizaron estos algoritmos en un esquema de clasificación jerárquica. En este esquema se plantean una serie de filtros que empiezan en la actividad más general (AMP) y terminan en una actividad especifica (Gram positivo o Gram negativo). Los resultados obtenidos muestran una fuerte tendencia por parte de los modelos obtenidos a tener valores muy altos de especificidad. Este comportamiento se debe principalmente al comportamiento del modelo al agregar clasificadores en serie, aunque también influye el tamaño de los conjuntos de entrenamientos en la capacidad de recuperación de los modelos.

Los resultados de la implementación directa de los clasificadores de una sola clase se muestran en la sección 4.2.1. Aquí observamos que estos modelos tienden a presentar una sensibilidad mucho más alta que sus valores de especificidad. Debido a que los modelos están entrenados únicamente con las instancias pertenecientes a la clase positiva (antimicrobial), era altamente probable que estos presentaran un valor de sensibilidad alta. Además, este es un comportamiento que se observa en la aplicación de estos algoritmos a otro tipo de problemas en la literatura como en los casos de Bezerra *et al.* (2019) y Alam *et al.* (2020). Por otro lado, los valores de especificidad relativamente bajos pueden tener dos explicaciones. Primero, la representación elegida hace imposible la separación total de ambos conjuntos. Segundo, debido a que el conjunto de prueba negativo no está validado experimentalmente, no podemos saber que realmente existe un porcentaje importante de muestras con actividad antimicrobiana en este conjunto. Es importante mencionar que a lo largo de la experimentación se utilizaron diferentes descriptores moleculares y se observó la misma tendencia del valor de especificidad, lo cual deja la segunda opción como la hipótesis más probable.

Utilizando la metodología descrita en el Capítulo 3, se obtuvieron los resultados presentados en la sección 4.2.3 y el Anexo B. Existe una gran variabilidad en estos resultados presentados, principalmente por la utilización de distintos descriptores moleculares para generar los modelos. A pesar de esto, podemos notar una marcada tendencia en estos modelos a presentar valores de especificidad muy altos con unos valores de sensibilidad relativamente más bajos, los cuales difieren de la utilización directa de los algoritmos de una sola clase, que presentan una proporción inversa a

estos valores. El comportamiento del modelo jerárquico, se debe principalmente al hecho de que en este se utilizan tres modelos de detección de anomalías en serie; el modelo AMP, antibacterial y ya sea el modelo Gram negativo o el Gram positivo, dependiendo del el flujo de detección que se utilice.

Al utilizar estos tres modelos conectados en serie, aumenta la probabilidad de que una muestra sea detectada como anomalía. Sea la probabilidad de que una muestra sea recuperada por el conjunto AMP P_{AMP}; la probabilidad de que la misma muestra sea recuperada por el conjunto antibacterial P_{AB}; y la probabilidad de que sea recuperada por el conjunto Gram P_{Gram}. En este caso podemos modelar la probabilidad de que una muestra sea recuperada por el modelo jerárquico como el producto de estas tres probabilidades:

$$P_{Total} = P_{AMP} \cdot P_{AB} \cdot P_{Gram} \tag{19}$$

Dado que en ninguno de los modelos presentados, se cuenta con una sensibilidad de 100 %, podemos deducir que para el caso de los modelos presentados, esta probabilidad de recuperación de una muestra siempre será menor al 100 %. Sin embargo, esta misma característica del modelo es la que permite que podamos obtener valores de especificidad relativamente altos, debido a que inicialmente, las muestras dadas al modelo para clasificar tienen una baja probabilidad de ser recuperadas por los tres modelos. Por lo tanto, podemos decir que aquellas muestras que sean recuperadas tienen una alta probabilidad de ser verdaderas muestras con la actividad biológica que se le está asignando.

Se puede observar una tendencia en estos modelos en la que una vez que llega a la etapa Gram positiva o Gram negativa, se presenta un decremento importante en la capacidad de recuperación de instancias positivas (sensibilidad). Se conjetura que esta pérdida de recuperación, está fuertemente ligada al tamaño de los conjuntos Gram positivo y Gram negativo. Estos conjuntos son mucho más reducidos si los comparamos con los conjuntos AMP o antibacterial, cuyos modelos muestran valores más altos de sensibilidad en todos los modelos. La razón por la cual no se ampliaron estos conjuntos, fue debido a la dificultad de obtener más muestras con actividad única, dado que estos conjuntos ya contienen todas las muestras con estas características que se encontraron en la base de datos starPep.

5.1.3. Comparación con los modelos de la literatura

Con el objetivo de comparar los resultados obtenidos por los modelos de clasificación de una sola clase con modelos binarios de la literatura, se planteó un experimento donde se clasificaron los conjuntos externos utilizando el mejor modelo AMP obtenido (AMP–ProtDcal-KME-EUC) y clasificadores binarios encontrados en la literatura. Estos resultados mostraron que el modelo AMP tiene una sensibilidad equiparable a la de estos modelos del estado del arte, sin embargo, esta tiende a tener una especificidad más pobre en comparación con los modelos binarios. Este fenómeno se debe a que, a diferencia de los modelos binarios, la etapa AMP está entrenada únicamente con ejemplos de muestras positivas, lo cual hace que sea más complicado para el modelo el diferenciar las muestras positivas de las negativas.

Existen limitaciones al comparar un nuevo modelo basado en QSAR, con otros modelos encontrados en literatura. En el caso de los clasificadores de péptidos antimicrobianos, incluso cuando se trata de comparar clasificadores binarios, hay que hacer consideraciones sobre la comparación que se presenta; dado que rara vez, estos modelos utilizan el mismo conjunto de entrenamiento, por lo tanto siempre tendrán diferentes dominios de aplicabilidad (ver sección 2.8), espacio de conocimiento, etc.

Al comparar modelos de clasificación de una sola clase con modelos binarios, estamos agregando una capa más de complejidad a la comparación, dado que al ser entrenados únicamente con la información de una sola clase, podríamos decir que estos algoritmos se encuentran de alguna manera en desventaja, contra los algoritmos binarios. Sin embargo, cuando se realizó una comparación directa, utilizando la metodología que se describe en la sección 4.3, se observó que la sensibilidad del modelo era superior a la de muchos de los modelos de la literatura, siendo únicamente superado por los modelos presentados por Pinacho-Castellanos *et al.* (2021b). En cuanto a los valores de especificidad presentados, aunque nuestro modelo es claramente inferior a la mayoría de los modelos presentados en la literatura, tenemos que tomar en consideración que el conjunto negativo de validación que se utilizó en esta prueba (conjunto Negativo Externo) no se encuentra validado, y por lo tanto solo puede tomarse como una referencia del verdadero valor de especificidad. Los modelos presentados en la literatura pueden obtener valores más altos de especificidad gracias a que están entrenados con conjuntos negativos que fueron obtenidos utilizando la metodología

que se explica en la sección 3.1.2 y por lo tanto, son capaces de reconocer estas muestras como negativas. Sin embargo, estas muestras podrían ser antimicrobianas y por lo tanto el desempeño de estos clasificadores estaría enmascarado por este fenómeno. Como ya se mencionó, la única forma de tener una verdadera validación de estos modelos es con un conjunto negativo experimentalmente validado.

También se realizó un experimento para observar el resultado de utilizar los algoritmos de una sola clase, en conjunto con los modelos de la literatura (clasificadores binarios). En esta experimentación pudimos observar que, en un flujo de clasificación jerárquica, el resultado tiende a perder un porcentaje de alrededor del 20 % de sensibilidad, en relación con lo obtenido por el modelo de la literatura; y también, tiende a ganar al rededor de un 5 % de especificidad. Más allá de discutir si es beneficiosa la utilización en conjunto de estos modelos particulares, lo que nos interesa resaltar, es que la pérdida de especificidad del modelo jerárquico indica que existen muestras que solo son recuperables por el modelo de una clase y por el binario, respectivamente. Por lo tanto, es posible que pueda encontrarse una metodología en que se optimicen los modelos para que estos puedan trabajar en conjunto y obtener mejores resultados. Esto último se plantea como una posible opción de trabajo futuro y se considera fuera del alcance de la presente investigación.

5.2. Conclusiones

En este trabajo se abordó el problema de clasificación de péptidos antimicrobianos utilizando algoritmos de clasificación de una clase. Se propuso una metodología de clasificación jerárquica basada en clasificadores de una clase, en la cual se buscó identificar péptidos Gram negativos y Gram positivos. Para esto, se generaron conjuntos de secuencias para las actividades: AMP, antibacteiral, Gram positivo y Gram negativo. Se describieron estos conjuntos utilizando descriptores de starPep, iFeature, ProDcal y BERT ESM. Después, se generaron modelos basados en estos descriptores, utilizando los algoritmos: “Bagging Isolation Forest” y “Bagging Bayes Net”. Al final, se evaluaron estos modelos utilizando las métricas de sensibilidad, especificidad y coeficiente de correlación de Matthews, y con base en esta última se determinó una jerarquía de estos modelos. A continuación se presentan las conclusiones del presente

trabajo de investigación.

1. La gran capacidad de recuperación de los modelos de una clase los hace ideales para su uso en situaciones donde se necesite encontrar la mayor cantidad de secuencias posible. Esto a diferencia de los modelos jerárquicos, donde su uso se recomienda en situaciones donde se desea minimizar la recuperación de falsos positivos.

2. Comparando diferentes descriptores y con base en la métrica del coeficiente de correlación de Matthews, los mejores modelos obtenidos son los que se generaron a partir de los descriptores de ProtDcal.

3. Comparando diferentes descriptores y con base en la métrica del coeficiente de correlación de Matthews, los peores modelos obtenidos son los que se generaron a partir de los descriptores de iFeature, debido a que estos obtienen los valores más bajos (negativos).

4. No se observa superioridad en modelos generados con la representación de BERT, en comparación con los modelos generados a partir de descriptores basados en propiedades fisicoquímicas (starPep y ProtDcal).

5. La unión de descriptores moleculares de starPep e iFeature produce modelos ligeramente superiores en sensibilidad a los generados únicamente con starPep. Este aumento de sensibilidad puede encontrarse entre el 1 % y 10 %. A su vez, se presenta un decremento en la especificidad en una proporción similar.

6. La etapa AMP del modelo generado con descriptores de ProtDcal, tiene una sensibilidad comparable o superior a la que con respecto a los modelos del estado del arte.

7. En el proceso de clasificación jerárquica, aumentar el intervalo de vecinos del algoritmo “Local Outlier Factor” no produce un cambio significativo en el resultado del modelo.

5.3. Trabajo futuro

Durante el desarrollo de la metodología y el proceso de experimentación se observaron oportunidades de investigación, que debido al limitado tiempo, no se tuvo oportunidad de abordar y por lo tanto estas quedaron fuera del alcance de esta investigación. A continuación se presenta una lista de estas posibles extensiones del presente trabajo.

1. **Conjunto negativo.** Con los resultados obtenidos por el proceso de clasificación de una clase, se formula una hipótesis en la que se sugiere que los conjuntos negativos de la Tabla 11 contienen muestras antimicrobianas. Para poder confirmar esta hipótesis es necesario realizar pruebas experimentales con las muestras de los conjuntos negativos que resultaran positivas después del proceso de clasificación.

2. **Selección de características.** La utilización de diferentes métodos de selección de características podría ayudar a generar mejores modelos. En esta investigación se utilizó la metodología de selección de características de la sección 3.3. Sin embargo, existen otras metodologías como las encontradas en Gu *et al.* (2011) y Jin *et al.* (2006).

3. **Optimización de hiperparámetros.** Se sugiere explorar metodologías para optimizar hiperparámetros, en los modelos de una clase, dado que al no tener un conjunto negativo validado, el único parámetro que puede optimizarse es la sensibilidad del modelo.

4. **Actividades biológicas.** En la presente investigación nos enfocamos únicamente a las actividades AMP, antibacterial, Gram positivo y Gram negativo. Sin embargo, la presente metodología puede extenderse a otras actividades biológicas como la antifúngica, antiviral o anticáncer.

Literatura citada

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

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

Aguilera-Mendoza, L., Marrero-Ponce, Y., García-Jacas, C. R., Chavez, E., Beltran, J. A., Guillen-Ramirez, H. A., y Brizuela, C. A. (2020). Automatic construction of molecular similarity networks for visual graph mining in chemical space of bioactive peptides: an unsupervised learning approach. *Scientific Reports*, **10**(18074).

Alam, S., Sonbhadra, S. K., Agarwal, S., y Nagabhushan, P. (2020). One-class support vector classifiers: A survey. *Knowledge-Based Systems*, **196**: 105754.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., y Watson, J. (2002). *Molecular Biology of the Cell. (4th ed.)*. Garland. pp. 130,142.

Alsini, R., Almakrab, A., Ibrahim, A., y Ma, X. (2021). Improving the outlier detection method in concrete mix design by combining the isolation forest and local outlier factor. *Construction and Building Materials*, **270**: 121396.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., y Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**(3): 403–410.

Andersson, D. I., Hughes, D., y Kubicek-Sutherland, J. Z. (2016). Mechanisms and consequences of bacterial resistance to antimicrobial peptides. *Drug Resistance Updates*, **26**: 43–57.

Barrett, P., Hunter, J., Miller, J. T., Hsu, J.-C., y Greenfield, P. (2005). matplotlib – A Portable Python Plotting Package. *ASP Conference Series*, **347**(June): 91.

Bateman, A. (2019). UniProt: A worldwide hub of protein knowledge. *Nucleic Acids Research*, **47**(D1): D506–D515.

Berg, J. M., Tymoczko, J. L., y Stryer, L. (2007). *Biochemistry (9th ed.)*. W. H. Freeman and Company 41. New York, pp. 65–106.

Bezerra, V. H., da Costa, V. G. T., Barbon Junior, S., Miani, R. S., y Zarpelão, B. B. (2019). IoTDS: A one-class classification approach to detect botnets in internet of things devices. *Sensors (Switzerland)*, **19**(14): 3188.

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

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, primera edición.

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

Breuniq, M. M., Kriegel, H. P., Ng, R. T., y Sander, J. (2000). LOF: Identifying density-based local outliers. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, **29**(2): 93–104.

Calvo, T., Kolesárová, A., Komorníková, M., y Mesiar, R. (2002). Aggregation Operators: Properties, Classes and Construction Methods. Springer, primera edición, capítulo Associativ, pp. 124–158.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., Webb, G. I., Smith, A. I., Daly, R. J., Chou, K. C., y Song, J. (2018). IFeature: A Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, **34**(14): 2499–2502.

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

Chou, K.-C. (2009). Pseudo Amino Acid Composition and its Applications in Bioinformatics, Proteomics and System Biology. *Current Proteomics*, **6**(4): 262–274.

Devlin, J., Chang, M. W., Lee, K., y Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. En: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, oct. Association for Computational Linguistics (ACL), Vol. 1, pp. 4171–4186.

Dubos, R. J. (1939). Studies on a bactericidal agent extracted from a soil bacillus : I. preparation of the agent. its activity in vitro. *The Journal of Experimental Medicine*, **70**(1): 1.

Epps, H. L. V. (2006). René Dubos: unearthing antibiotics. *The Journal of Experimental Medicine*, **203**(2): 259.

Evgeniou, T. y Pontil, M. (2001). Support vector machines: Theory and applications. 01. Vol. 2049, pp. 249–257.

Fix, E. y Hodges, J. L. (1989). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *International Statistical Review / Revue Internationale de Statistique*, **57**(3): 238.

Gu, Q., Li, Z., y Han, J. (2011). Generalized fisher score for feature selection. En: *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*. pp. 266–273.

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

Hanser, T., Barber, C., Guesné, S., Marchaland, J. F., y Werner, S. (2019). Applicability Domain: Towards a More Formal Framework to Express the Applicability of a Model and the Confidence in Individual Predictions. *Challenges and Advances in Computational Chemistry and Physics*, **30**: 215–232.

Heckerman, D. (1997). Bayesian Networks for Data Mining. *Data Mining and Knowledge Discovery 1997 1:1*, **1**(1): 79–119.

Hoaglin, D. C., Iglewicz, B., y Tukey, J. W. (1986). Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association*, **81**(396): 991–999.

Irigoien, I., Sierra, B., y Arenas, C. (2014). Towards application of one-class classification methods to medical data. *TheScientificWorldJournal*, **2014**: 730712.

J, Y., P, B., A, L., P, S., L, Q., HK, T., KH, W., y SWI, S. (2020). Deep-AmPEP30: Improve Short Antimicrobial Peptides Prediction with Deep Learning. *Molecular therapy. Nucleic acids*, **20**: 882–894.

Jhong, J. H., Chi, Y. H., Li, W. C., Lin, T. H., Huang, K. Y., y Lee, T. Y. (2019). dbAMP: an integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data. *Nucleic Acids Research*, **47**(D1): D285–D297.

Jin, X., Xu, A., Bie, R., y Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **3916 LNBI**: 106–115.

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

Kang, X., Dong, F., Shi, C., Liu, S., Sun, J., Chen, J., Li, H., Xu, H., Lao, X., y Zheng, H. (2019). DRAMP 2.0, an updated data repository of antimicrobial peptides. *Scientific Data 2019 6:1*, **6**(1): 1–10.

Kumar, V. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, **4**(3): 211–229.

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

Li, J., Koh, J. J., Liu, S., Lakshminarayanan, R., Verma, C. S., y Beuerman, R. W. (2017). Membrane active antimicrobial peptides: Translating mechanistic insights to design. **11**(FEB): 73.

Li, W. y Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**(13): 1658–1659.

Lin, Y., Cai, Y., Liu, J., Lin, C., y Liu, X. (2019). An advanced approach to identify antimicrobial peptides and their function types for penaeus through machine learning strategies. *BMC Bioinformatics*, **20**(8): 1–10.

Liu, F. T., Ting, K. M., y Zhou, Z. H. (2008). Isolation forest. En: *Proceedings - IEEE International Conference on Data Mining, ICDM*. pp. 413 – 422.

McLachlan, S., Dube, K., Hitman, G. A., Fenton, N. E., y Kyrimi, E. (2020). Bayesian networks in healthcare: Distribution by medical condition. *Artificial Intelligence in Medicine*, **107**: 101912.

Mehta, D., Anand, P., Kumar, V., Joshi, A., Mathur, D., Singh, S., Tuknait, A., Chaudhary, K., Gautam, S. K., Gautam, A., Varshney, G. C., y Raghava, G. P. (2014). ParaPep: A Web resource for experimentally validated antiparasitic peptide sequences and their structures. *Database*, **2014**: 1–7.

Miller, D., Sunderhauf, N., Milford, M., y Dayoub, F. (2021). Class anchor clustering: A loss for distance-based open set recognition. En: *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*. pp. 3569–3577.

Nantasenamat, C. (2020). Best practices for constructing reproducible QSAR models. *Methods in Pharmacology and Toxicology*, pp. 55–75.

Nelson, D. L. y Cox, M. M. (2008). *Principles of Biochemistry (5th ed.)*. W.H. Freeman Company. pp. 142–170.

Ng, X., Rosdi, B., y Shahrudin, S. (2015). Prediction of antimicrobial peptides based on sequence alignment and support vector machine-pairwise algorithm utilizing lz-complexity. *BioMed research international*, **2015**: 212715.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., y Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**(2): 2825–2830.

Perera, P., Oza, P., y Patel, V. M. (2021). One-Class Classification: A Survey. *arXiv preprint arXiv:2101.03064*, **6**(10).

Peters, B. M., Shirtliff, M. E., y Jabra-Rizk, M. A. (2010). Antimicrobial Peptides: Primeval Molecules or Future Drugs? *PLOS Pathogens*, **6**(10): e1001067.

Phoenix, D. A., Dennison, S. R., y Harris, F. (2013). Antimicrobial Peptides: Their History, Evolution, and Functional Promiscuity. *Antimicrobial Peptides*, pp. 1–37.

Pinacho-Castellanos, S. A., García-Jacas, C. R., Gilson, M. K., y Brizuela, C. A. (2021a). Alignment-Free Antimicrobial Peptide Predictors: Improving Performance by a Thorough Analysis of the Largest Available Data Set. *Journal of Chemical Information and Modeling*, **61**(6): 3141–3157.

Pinacho-Castellanos, S. A., García-Jacas, C. R., Gilson, M. K., y Brizuela, C. A. (2021b). Alignment-Free Antimicrobial Peptide Predictors: Improving Performance by a Thorough Analysis of the Largest Available Data Set. *Journal of Chemical Information and Modeling*, **61**(6): 3141–3157.

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.

Pirtskhalava, M., Gabrielian, A., Cruz, P., Griggs, H. L., Squires, R. B., Hurt, D. E., Grigolava, M., Chubinidze, M., Gogoladze, G., Vishnepolsky, B., Alekseev, V., Rosenthal, A., y Tartakovsky, M. (2016). DBAASP v.2: An enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides. *Nucleic Acids Research*, **44**(D1): D1104–12.

Qureshi, A., Thakur, N., Tandon, H., y Kumar, M. (2014). AVPdb: A database of experimentally validated antiviral peptides targeting medically important viruses. *Nucleic Acids Research*, **42**(D1): D1147–D1153.

Richon, A. y Young, S. (1997). An introduction to QSAR methodology. Recuperado: junio de 2021, de: http://www.netsci.org/Science/Compchem/feature19.html.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., y Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, **118**(15): 622803.

Robitaille, T., Beaumont, C., Qian, P., Borkin, M., y Goodman, A. (2019). glueviz v0.15.2: multidimensional data exploration. Recuperado: junio de 2021, de: https://zenodo.org/record/3385920.

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

Roy, K., Kar, S., y Das, R. N. (2015). *A Primer on QSAR/QSPR Modeling: Fundamental Concepts*. Springer, Cham, primera edición. Heidelberg, pp. 10–62.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, **27**(3): 379–423.

SJ, K., DH, K., T, M.-O., y BJ, L. (2012). Antimicrobial peptides: their physicochemical properties and therapeutic application. *Archives of pharmacal research*, **35**(3): 409–413.

Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K., y Idicula-Thomas, S. (2010). CAMP: a useful resource for research on antimicrobial peptides. *Nucleic Acids Research*, **38**(1): D774–D780.

Todeschini, R. y Consonni, V. (2009). *Molecular Descriptors for Chemoinformatics: Volume I: Alphabetical Listing / Volume II: Appendices, References, Volume 41*. John Wiley Sons, Ltd, primera edición. Raimund Mannhold, Hugo Kubinyi, Gerd Folkers, pp. 15–19.

Torrent, M., Nogués, V. M., y Boix, E. (2009). A theoretical approach to spot active regions in antimicrobial proteins. *BMC Bioinformatics*, **10**(1): 1–9.

Torrent, M., Di Tommaso, P., Pulido, D., Nogués, M. V., Notredame, C., Boix, E., y Andreu, D. (2012). AMPA: an automated web server for prediction of protein antimicrobial regions. *Bioinformatics*, **28**(1): 130–131.

Tortora, G., Funke, B., Case, C., Weber, D., y Bair, W. (2019). *Microbiology: An Introduction*. Pearson. pp. 135–179.

Tyagi, A., Tuknait, A., Anand, P., Gupta, S., Sharma, M., Mathur, D., Joshi, A., Singh, S., Gautam, A., y Raghava, G. P. (2015). CancerPPD: a database of anticancer peptides and proteins. *Nucleic Acids Research*, **43**(D1): D837–D843.

Tyagi, A., Pankaj, V., Singh, S., Roy, S., Semwal, M., Shasany, A. K., y Sharma, A. (2019). PlantAFP: a curated database of plant-origin antifungal peptides. *Amino Acids 2019 51:10*, **51**(10): 1561–1568.

Van Rossum, G. y Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace. Scotts Valley, CA, pp. 5–10.

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

Vijayakumar, V., Divya, N. S., Sarojini, P., y Sonika, K. (2020). Isolation Forest and Local Outlier Factor for Credit Card Fraud Detection System. *International Journal of Engineering and Advanced Technology*, **9**(4): 261–265.

Vriza, A., Canaj, A. B., Vismara, R., Cook, L. J. K., Manning, T. D., Gaultois, M. W., Wood, P. A., Kurlin, V., Berry, N., Dyer, M. S., y Rosseinsky, M. J. (2021). One class classification as a practical approach for accelerating π–π co-crystal discovery. *Chemical Science*, **12**(5): 1702–1719.

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.

WHO (2014). Antimicrobial resistance. Global report on surveillance. *World Health Organization*. Recuperado: agosto de 2021, de: https://apps.who.int/iris/ handle/10665/112642.

Willey, J. M., Prescott, L. M., Sandman, K. M., y Wood, D. H. D. H. (2020). *Prescott's microbiology (7th ed.)*. Oxford Academic. pp. 100–114.

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.

Zar, J. H. (2005). Spearman rank correlation. En: *Encyclopedia of Biostatistics*. John Wiley Sons, Ltd.

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

Anexo A - Conjuntos de entrenamiento y validación utilizados

Tabla 26. Conjuntos de péptidos utilizados en la experimentación.

Archivo	Actividad	Función	Tamaño (secuencias)
AMP_TR.fasta	antibacterial, antiparasitaria, antiviral, antifúngico	Entrenamiento	8803
Antibacterial_TR.fasta	antibacterial	Entrenamiento	4844
Gram_neg_TR.fasta	Gram negativo	Entrenamiento	572
Gram_pos_TR.fasta	Gram positivo	Entrenamiento	375
Gram_neg_TS.fasta	Gram negativo	Validación	144
Gram_pos_TS.fasta	Gram positivo	Validación	94
Gram_EXT_TS.fasta	antibacterial	Validación	5856
Neg_TS.fasta	negativo no validado	Validación	1695
Neg_EXT_TS.fasta	negativo no validado	Validación	10771

Tabla 27. Enlaces de descarga de los conjuntos.

Archivo	Enlace Descarga
AMP_TR.fasta	https://drive.google.com/file/d/1ICqf3XuiAZf7j14DEHUGbCMGAGXl78Qu/view?usp=sharing
Antibacterial_TR.fasta	https://drive.google.com/file/d/1SJJ4Y5bJ4C8IMndYPSVOnu2KYPJJq72Y/view?usp=sharing
Gram_neg_TR.fasta	https://drive.google.com/file/d/1E0NAfktNQli3-KiNRCaAeKm4gYc1t9iv/view?usp=sharing
Gram_pos_TR.fasta	https://drive.google.com/file/d/1OAr5nBq9F3qRw4ER329RbdHfskozdnIp/view?usp=sharing
Gram_neg_TS.fasta	https://drive.google.com/file/d/1Zjn6pn96rKG8Jpomipz4nWVSMVm9sy-A/view?usp=sharing
Gram_pos_TS.fasta	https://drive.google.com/file/d/1X-8qN-ytupFM1PA7mQSY5ATGfLuVHyxl/view?usp=sharing
Gram_EXT_TS.fasta	https://drive.google.com/file/d/14Ukl6oIJHZxL96v-Ld9LSd6WbdWfhMff/view?usp=sharing
Neg_TS.fasta	https://drive.google.com/file/d/16z4hHPzbU6-pW1krR_ugvMR58EJMvsh_/view?usp=sharing
Neg_EXT_TS.fasta	https://drive.google.com/file/d/1dDXQWFvjMXRILPoqhy4iN5Qsl1nQAHwh/view?usp=sharing

Anexo B - Resultados de experimentación

Tabla 28. Resultados del modelo construido con descriptores starPep para el conjunto Gram Negativo.

Identificador:		Starpep #1											
Prueba:		Gram negativo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		93		97		52							
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	68.75	66.66	27.77	19.44	68.05	70.13	36.11	28.47	66.66	72.22	41.66	33.33
	Outlier	63.59	74.35	70.38	88.37	69.55	73.98	85.72	94.86	67.49	71.09	78.82	91.26
	MCC	0.17	0.24	-0.01	0.06	0.21	0.25	0.16	0.24	0.19	0.24	0.13	0.21
Chebyshev	Target	72.91	64.58	30.55	22.22	75.69	70.83	29.86	25.0	76.38	74.30	50.69	43.05
	Outlier	55.75	74.45	81.00	90.02	61.17	73.68	95.16	97.34	58.99	70.14	74.86	87.84
	MCC	0.15	0.23	0.07	0.10	0.20	0.26	0.26	0.29	0.19	0.25	0.15	0.23
Manhattan	Target	72.91	78.47	32.63	27.77	70.83	83.33	33.33	30.55	72.91	83.33	37.5	31.94
	Outlier	60.47	62.71	67.78	78.64	64.36	62.53	90.73	94.27	61.59	58.93	78.70	86.96
	MCC	0.18	0.22	0.002	0.04	0.19	0.25	0.20	0.25	0.18	0.22	0.10	0.14

Mejores Modelos:	AMP	Target: KMM - CHB	76.38	AB	Target: KME - MHT	83.33	Gram	Target: KMM - CHB	50.69
		Outlier: KME -EUC	69.55		Outlier: EM -CHB	74.45		Outlier: KME -CHB	95.16
		MCC: KME -EUC	0.21		MCC: KME -CHB	0.26		MCC: KME -CHB	0.26

Tabla 29. Resultados del modelo construido con descriptores starPep para el conjunto Gram Positivo.

Identificador:		Starpep #1											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		93		97		85							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	50.0	46.80	32.97	19.14	52.12	48.93	37.23	28.72	55.31	56.38	38.29	32.97
	Outlier	63.59	74.39	92.09	96.04	69.55	73.98	85.89	93.45	67.49	71.09	87.37	92.50
	MCC	0.06	0.10	0.19	0.15	0.10	0.11	0.14	0.18	0.10	0.13	0.16	0.20
Chebyshev	Target	59.57	46.80	13.82	10.63	58.51	53.19	18.08	14.89	64.89	56.38	21.27	17.02
	Outlier	55.75	74.45	99.76	99.76	61.17	73.68	98.99	99.23	58.99	70.14	97.75	98.64
	MCC	0.06	0.10	0.31	0.26	0.08	0.13	0.27	0.25	0.10	0.12	0.23	0.23
Manhattan	Target	67.02	65.95	26.59	26.59	58.51	65.95	22.34	20.21	61.70	65.95	21.27	19.14
	Outlier	60.47	62.71	96.63	97.22	64.36	62.53	97.64	98.28	61.59	58.93	97.10	97.64
	MCC	0.12	0.13	0.24	0.27	0.10	0.13	0.24	0.25	0.10	0.11	0.21	0.21

Mejores Modelos:	AMP	Target: KMM - CHB	67.02	AB	Target: KME - MHT	65.95	Gram	Target: KMM - CHB	38.29
		Outlier: KME -EUC	69.55		Outlier: EM -CHB	74.45		Outlier: KME -CHB	96.76
		MCC: KME -EUC	0.12		MCC: KME -CHB	0.13		MCC: KME -CHB	0.31

Tabla 30. Resultados del modelo construido con descriptores starPep para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		Starpep #1											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		93		97		52							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	66.68	58.50	26.84	17.69	67.64	58.84	22.13	17.23	69.29	65.55	18.37	14.34
	Outlier	62.07	77.52	58.12	87.53	69.54	75.87	75.23	92.70	67.30	71.53	66.62	88.75
	MCC	0.27	0.35	-0.14	0.07	0.35	0.34	-0.02	0.15	0.34	0.35	-0.15	0.04
Chebyshev	Target	76.22	58.60	18.61	13.88	76.55	58.84	16.97	12.85	77.51	64.90	27.27	22.37
	Outlier	56.96	74.60	77.64	89.47	62.06	74.14	83.30	91.73	58.45	71.43	66.89	85.60
	MCC	0.31	0.32	-0.04	0.04	0.036	0.32	0.00	0.07	0.33	0.34	-0.05	0.09
Manhattan	Target	73.12	73.08	28.56	24.26	73.56	74.65	23.99	23.01	74.98	77.23	33.84	31.25
	Outlier	58.17	65.11	61.07	77.79	62.44	64.70	84.91	92.06	60.84	61.60	67.44	83.51
	MCC	0.29	0.35	-0.10	0.02	0.33	0.36	0.10	0.21	0.33	0.36	0.01	0.16

Mejores Modelos:	AMP	Target: KMM - CHB	77.51	AB	Target: KMM - MHT	77.23	Gram	Target: KMM - MHT	33.84
		Outlier: KME -EUC	69.54		Outlier: EM -EUC	77.52		Outlier: KME -MHT	84.91
		MCC: KME -CHB	0.36		MCC: KME-MHT	0.36		MCC: KME -MHT	0.10

Tabla 31. Resultados del modelo construido con descriptores starPep para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		Starpep #1			
Prueba:		Gram positivo - Ext			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram positivo (375)	
Numero de descriptores:		93	97	85	
Conjuntos de prueba:		Positivo: Gram_Ext (5856)		Negativo: Ext_Neg (10771)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	66.68	58.50	18.81	13.64	67.64	58.84	32.44	24.02	69.29	65.55	28.43	24.36
	Outlier	62.07	77.52	86.54	94.44	69.54	75.84	81.33	91.66	67.30	71.53	90.16	89.95
	MCC	0.27	0.35	0.07	0.13	0.35	0.34	0.15	0.21	0.34	0.35	0.09	0.18
Chebyshev	Target	76.22	58.60	5.75	5.29	76.55	58.84	9.27	8.19	77.51	64.90	7.13	6.69
	Outlier	56.96	74.60	99.46	99.60	62.06	74.14	97.60	98.03	58.45	71.43	94.70	97.10
	MCC	0.31	0.32	0.16	0.16	0.36	0.32	0.15	0.14	0.33	0.34	0.03	0.08
Manhattan	Target	73.12	73.08	10.96	10.16	75.56	74.65	9.01	8.53	74.98	77.23	10.39	9.83
	Outlier	58.17	65.11	93.66	95.17	62.44	64.70	95.17	96.42	60.84	61.60	94.22	95.91
	MCC	0.29	0.35	0.08	0.10	0.33	0.36	0.08	0.10	0.33	0.36	0.08	0.11

Mejores Modelos:	AMP	Target: KMM - CHB	77.51	AB	Target: KMM - MHT	77.23	Gram	Target: KME - EUC	32.44
		Outlier: KME -EUC	69.54		Outlier: EM -EUC	77.52		Outlier: EM -CHB	99.46
		MCC: KME -CHB	0.36		MCC: KME -MHT	0.36		MCC: EM -CHB	0.16

Tabla 32. Resultados del modelo construido con descriptores starPep para el conjunto Gram Negativo.

Identificador:		Starpep #2			
Prueba:		Gram negativo			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram negativo (572)	
Numero de descriptores:		111	104	52	
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)		Negativo: AB_Neg (1695)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	78.47	62.5	18.05	13.88	74.30	63.88	21.52	17.36	73.61	61.80	25.69	20.13
	Outlier	63.06	79.94	99.23	99.94	65.78	84.07	96.81	98.40	64.66	80.53	94.98	97.93
	MCC	0.22	0.26	0.32	0.34	0.22	0.32	0.23	0.25	0.21	0.27	0.22	0.26
Chebyshev	Target	72.91	66.66	13.19	10.41	73.61	70.83	25.0	21.52	75.0	72.22	27.08	22.22
	Outlier	62.77	68.67	99.70	99.88	64.01	77.10	98.34	98.93	63.53	75.98	99.76	99.82
	MCC	0.19	0.20	0.30	0.28	0.20	0.29	0.34	0.34	0.21	0.28	0.47	0.43
Manhattan	Target	82.63	70.83	12.5	9.72	72.22	71.52	18.05	15.27	76.38	74.34	23.61	20.13
	Outlier	56.99	68.67	99.05	99.46	62.18	75.92	99.76	99.82	59.0	70.79	99.58	99.64
	MCC	0.21	0.22	0.23	0.22	0.18	0.28	0.37	0.35	0.19	0.25	0.42	0.38

Mejores Modelos:	AMP	Target: EM - MHT	82.63	AB	Target: KMM - MHT	74.34	Gram	Target: KMM -CHB	27.08
		Outlier: KME -EUC	65.78		Outlier: KME -EUC	84.07		Outlier: KMM -CHB	99.76
		MCC: EM-EUC	0.22		MCC: KME -EUC	0.32		MCC: KMM -CHB	0.47

Tabla 33. Resultados del modelo construido con descriptores starPep para el conjunto Gram Positivo.

Identificador:		Starpep #2											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		111		104		79							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	65.95	51.06	28.72	22.34	57.44	45.74	17.02	14.89	59.57	48.93	20.21	19.14
	Outlier	63.06	79.94	92.80	96.69	65.78	84.07	99.64	99.64	64.60	80.53	99.17	99.52
	MCC	0.13	0.16	0.17	0.20	0.10	0.17	0.33	0.30	0.11	0.16	0.32	0.34
Chebyshev	Target	58.51	57.44	24.46	22.34	53.19	51.06	26.59	22.34	53.19	51.06	24.46	20.21
	Outlier	62.77	68.67	96.93	97.81	64.01	77.10	97.40	98.40	63.53	75.98	96.10	98.05
	MCC	0.09	0.12	0.23	0.25	0.07	0.14	0.27	0.28	0.07	0.13	0.21	0.24
Manhattan	Target	61.70	56.38	26.59	18.08	63.82	56.38	23.40	21.27	58.51	61.70	25.53	21.27
	Outlier	56.99	68.67	90.50	93.74	62.18	75.92	95.64	96.99	59.70	70.79	97.22	98.05
	MCC	0.08	0.11	0.12	0.10	0.11	0.16	0.18	0.20	0.08	0.15	0.26	0.25

Mejores Modelos:	AMP	Target: EM - EUC	65.95	AB	Target: KMM - MHT	61.70	Gram	Target: EM-EUC	28.72
		Outlier: KME -EUC	65.78		Outlier: KME -EUC	84.07		Outlier: KME -EUC	99.64
		MCC: EM-EUC	0.13		MCC: KME -EUC	0.17		MCC: KME -EUC	0.33

Tabla 34. Resultados del modelo construido con descriptores starPep para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		Starpep #2											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		111		104		52							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	75.56	53.62	6.96	5.68	73.65	53.17	8.26	6.19	74.94	53.97	12.17	9.11
	Outlier	64.51	79.55	98.18	99.43	68.07	84.02	94.95	97.82	66.32	84.17	91.02	97.56
	MCC	0.35	0.33	0.13	0.16	0.39	0.38	0.06	0.10	0.38	0.39	0.05	0.14
Chebyshev	Target	73.05	62.32	2.39	1.87	74.07	62.61	3.87	3.60	75.52	63.79	4.08	3.65
	Outlier	63.24	69.86	99.49	99.61	66.17	79.44	98.18	98.75	66.08	76.83	99.66	99.77
	MCC	0.34	0.30	0.08	0.07	0.37	0.41	0.06	0.07	0.39	0.39	0.14	0.13
Manhattan	Target	76.19	66.78	5.14	4.83	77.33	63.46	4.08	3.89	77.33	67.48	3.80	3.67
	Outlier	57.91	71.34	98.11	98.78	64.51	76.37	99.65	99.76	61.45	71.88	99.24	99.43
	MCC	0.32	0.36	0.09	0.11	0.39	0.41	0.14	0.14	0.36	0.37	0.10	0.11

Mejores Modelos:	AMP	Target: KME - MHT	77.33	AB	Target: KMM - MHT	67.48	Gram	Target: KMM-MHT	12.17
		Outlier: KME -EUC	68.07		Outlier: KMM -EUC	84.17		Outlier: KMM -CHB	99.66
		MCC: KME-MHT	0.39		MCC: KME -CHB	0.41		MCC: KMM -CHB	0.14

Tabla 35. Resultados del modelo construido con descriptores starPep para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		Starpep #2											
Prueba:		Gram positivo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		111		104		79							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	73.56	53.62	14.73	11.08	73.65	53.17	5.65	4.95	74.94	53.97	11.32	8.86
	Outlier	64.51	79.55	87.76	94.83	68.07	84.02	99.50	99.63	66.32	84.17	98.39	99.33
	MCC	0.35	0.33	00.3	0.10	0.39	0.38	0.16	0.15	0.38	0.39	0.21	0.21
Chebyshev	Target	73.05	62.32	12.22	10.80	74.07	62.61	12.99	11.28	75.52	63.79	16.17	14.10
	Outlier	63.24	69.86	93.68	95.85	66.17	79.44	96.61	98.14	66.08	76.83	96.54	97.85
	MCC	0.34	0.30	0.10	0.12	0.37	0.41	0.18	0.20	0.39	0.39	0.22	0.23
Manhattan	Target	76.19	66.78	16.66	14.42	77.33	63.46	14.83	13.13	77.33	67.48	10.38	9.47
	Outlier	57.91	71.34	84.22	91.27	64.51	76.37	92.36	95.50	61.45	71.85	94.71	96.45
	MCC	0.32	0.36	0.01	0.08	0.39	0.41	0.11	0.15	0.36	0.37	0.09	0.12

Mejores Modelos:	AMP	Target: KME - MHT	77.33	AB	Target: KMM - MHT	67.48	Gram	Target: EM-MHT	16.66
		Outlier: KME -EUC	68.07		Outlier: KMM -EUC	84.17		Outlier: KME -EUC	99.50
		MMC: KME-MHT	0.39		MCC: KME -CHB	0.41		MCC: KMM-CHB	0.22

Tabla 36. Resultados del modelo construido con descriptores iFeature para el conjunto Gram negativo.

Identificador:		iFeature											
Prueba:		Gram negativo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		35		34		27							
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	76.92	66.43	45.45	35.66	71.32	80.41	53.84	49.65	77.62	79.72	54.54	51.04
	Outlier	9.50	14.22	24.20	30.04	16.23	14.40	16.76	26.32	13.16	11.62	25.50	29.33
	MCC	-0.11	-0.14	-0.18	-0.19	-0.08	-0.03	-0.20	-0.14	-0.07	-0.07	-0.12	-0.11
Chebyshev	Target	76.92	63.63	29.37	23.07	74.12	80.41	46.15	44.75	76.22	81.11	53.14	50.34
	Outlier	9.50	17.06	33.53	39.13	13.28	18.89	24.38	31.70	10.56	14.93	24.49	28.57
	MCC	-0.11	-0.13	-0.20	-0.20	-0.09	-0.00	-0.17	-0.13	-0.11	-0.2	-0.13	-0.12
Manhattan	Target	77.62	66.43	34.96	30.06	67.13	76.92	41.95	41.25	74.12	80.41	41.25	39.16
	Outlier	14.34	16.88	27.09	33.76	15.64	16.41	26.03	30.87	12.63	13.63	26.74	29.39
	MCC	-0.06	-0.11	-0.22	-0.20	-0.12	-0.04	-0.18	-0.15	-0.10	-0.04	-0.18	-0.18

Mejores Modelos:	AMP	Target: EM - MHT	77.62	AB	Target: KMM -CHB	81.11	Gram	Target: KMM-EUC	54.54
		Outlier: KME -EUC	16.23		Outlier: KME-CHB	18.89		Outlier: EM-CHB	33.53
		MCC: EM-MHT	-0.06		MCC: KME -CHB	-0.004		MCC: KMM-EUC	-0.12

Tabla 37. Resultados del modelo construido con descriptores iFeature para el conjunto Gram positivo.

Identificador:		iFeature			
Prueba:		Gram positivo			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram positivo (375)	
Numero de descriptores:		35	34	33	
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)		Negativo: AB_Neg (1695)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	79.56	67.74	64.51	50.53	66.66	75.26	77.41	55.91	76.34	78.49	75.26	64.51
	Outlier	9.50	14.22	16.46	23.78	16.23	14.40	7.43	21.42	13.16	11.62	9.20	17.94
	MCC	-0.08	-0.11	-0.11	-0.13	-0.10	-0.06	-0.12	-0.12	-0.06	-0.06	-0.11	-0.09
Chebyshev	Target	79.56	66.66	44.08	36.55	73.11	69.89	48.38	40.86	77.41	68.81	54.83	48.38
	Outlier	9.50	17.06	31.99	36.18	13.28	18.89	33.23	37.24	10.56	14.93	22.55	27.33
	MCC	-0.08	-0.09	-0.11	-0.12	-0.08	-0.06	-0.08	-0.09	-0.08	-0.08	-0.11	-0.11
Manhattan	Target	76.34	64.51	39.78	32.25	69.89	70.96	43.01	39.78	78.49	73.11	47.31	44.08
	Outlier	14.34	16.88	26.74	34.59	15.64	16.41	38.25	40.79	12.63	13.63	28.15	30.99
	MCC	-0.05	-0.10	-0.16	-0.15	-0.08	-0.07	-0.08	-0.08	-0.05	-0.08	-0.11	-0.11

Mejores Modelos:	AMP	Target: EM - EUC	79.56	AB	Target: KMM -EUC	78.49	Gram	Target: KME-EUC	77.41
		Outlier: KME-EUC	16.23		Outlier: KME-CHB	18.89		Outlier: KME-MHT	38.25
		MCC: KMM-MHT	-0.05		MCC: KMM-EUC	-0.06		MCC: KME-MHT	-0.08

Tabla 38. Resultados del modelo construido con descriptores iFeature para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		iFeature			
Prueba:		Gram negativo - Ext			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram negativo (572)	
Numero de descriptores:		35	34	27	
Conjuntos de prueba:		Positivo: Gram_Ext (5856)		Negativo: Ext_Neg (10771)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	74.07	72.40	38.76	25.59	74.81	77.28	42.58	31.86	76.99	78.05	43.15	34.35
	Outlier	8.71	7.26	16.04	25.87	7.92	6.51	9.72	19.17	5.63	4.75	16.12	20.98
	MCC	-0.23	-0.27	-0.46	-0.46	-0.23	-0.23	-0.51	-0.48	-0.25	-0.26	-0.42	-0.44
Chebyshev	Target	76.79	68.40	29.30	19.34	77.03	74.31	28.92	21.56	79.35	75.37	40.45	31.31
	Outlier	4.52	10.97	23.63	32.42	5.74	9.18	15.53	24.07	4.12	6.43	15.38	21.06
	MCC	-0.28	-0.25	-0.45	-0.46	-0.25	-0.22	-0.55	-0.52	-0.26	-0.25	-0.45	-0.46
Manhattan	Target	73.37	68.51	34.76	22.96	73.95	73.95	32.71	24.41	76.51	76.12	41.82	32.37
	Outlier	7.33	9.91	16.65	26.67	7.47	8.03	15.32	23.08	5.50	5.54	18.25	23.34
	MCC	-0.26	-0.27	-0.49	-0.48	-0.25	-0.24	-0.52	-0.50	-0.26	-0.26	-0.40	-0.43

Mejores Modelos:	AMP	Target: KMM - CHB	79.35	AB	Target: KMM -EUC	78.05	Gram	Target: KMM-EUC	43.15
		Outlier: EM-EUC	8.71		Outlier: EM-CHB	10.97		Outlier: EM-CHB	23.63
		MCC: EM-EUC	-0.23		MCC: KME-CHB	-0.22		MCC: KMM-MHT	-0.40

Tabla 39. Resultados del modelo construido con descriptores iFeature para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		iFeature											
Prueba:		Gram positivo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		35		34		33							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	74.07	72.40	60.89	40.36	74.81	77.28	72.79	51.81	76.99	78.05	73.83	54.91
	Outlier	8.71	7.26	9.82	20.71	7.92	6.51	3.84	15.21	5.63	4.75	3.99	11.54
	MCC	-0.23	-0.27	-0.34	-0.39	-0.23	-0.23	-0.34	-0.35	-0.25	-0.26	-0.32	-0.37
Chebyshev	Target	76.79	68.40	46.20	31.35	77.03	74.31	39.46	29.93	79.35	75.37	47.42	36.30
	Outlier	4.52	10.97	22.54	31.52	5.74	9.18	25.00	31.96	4.12	6.43	13.74	19.86
	MCC	-0.28	-0.25	-0.31	-0.35	-0.25	-0.22	-0.35	-0.36	-0.26	-0.25	-0.41	-0.43
Manhattan	Target	73.37	68.51	44.82	30.03	73.95	73.95	33.38	25.44	76.51	76.12	41.64	32.34
	Outlier	7.33	9.91	20.28	30.40	7.47	8.03	29.41	35.33	5.50	5.54	18.07	23.29
	MCC	-0.26	-0.27	-0.35	-0.38	-0.25	-0.24	-0.35	-0.37	-0.26	-0.26	-0.41	-0.43

Mejores Modelos:	AMP	Target: KMM - CHB	79.35	AB	Target: KMM -EUC	78.05	Gram	Target: KMM-EUC	73.83
		Outlier: EM-EUC	8.71		Outlier: EM-CHB	10.97		Outlier: KME-MHT	29.41
		MCC: EM-EUC	-0.23		MCC: KME-CHB	-0.22		MCC: EM-CHB	-0.31

Tabla 40. Resultados del modelo construido con descriptores starPep + iFeature #1 para el conjunto Gram negativo.

Identificador:		Starpep + iFeature #1											
Prueba:		Gram negativo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		128		131		79							
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	77.77	76.38	17.36	15.27	74.30	78.47	40.27	37.5	73.61	79.16	35.41	34.02
	Outlier	47.02	61.41	81.71	91.38	52.21	62.59	75.63	89.55	48.49	60.0	64.48	83.12
	MCC	0.13	0.20	-0.00	0.06	0.14	0.22	0.09	0.21	0.11	0.21	-0.00	0.11
Chebyshev	Target	72.91	68.75	16.66	14.58	73.61	78.47	26.38	25.69	74.30	73.61	23.61	21.52
	Outlier	51.09	56.28	74.98	87.78	53.56	64.48	75.69	90.20	51.38	61.59	86.25	95.22
	MCC	0.12	0.13	-0.05	0.01	0.14	0.23	0.01	0.13	0.13	0.19	0.07	0.18
Manhattan	Target	72.22	74.30	29.16	25.69	73.61	79.16	43.75	39.58	76.38	76.38	47.22	43.05
	Outlier	48.43	58.40	48.90	70.44	54.21	60.58	64.95	80.35	50.44	55.39	59.82	74.45
	MCC	0.11	0.17	-0.11	-0.02	0.14	0.21	0.04	0.13	0.14	0.17	0.03	0.10

Mejores Modelos:	AMP	Target: EM-EUC	77.77	AB	Target: KME-MHT	79.16	Gram	Target: KMM-MHT	47.22
		Outlier: KME-MHT	54.21		Outlier: KME-CHB	64.48		Outlier: KMM-CHB	86.25
		MCC: KME-MHT	0.14		MCC: KME-CHB	0.23		MCC: KME-EUC	0.09

Tabla 41. Resultados del modelo construido con descriptores starPep + iFeature #1 para el conjunto Gram positivo.

Identificador:		Starpep + iFeature #1											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		128		131		118							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	71.27	67.02	34.04	30.85	65.95	63.82	37.23	30.85	71.27	63.82	28.72	28.72
	Outlier	47.02	61.41	83.59	89.26	52.21	62.59	79.46	85.95	48.49	60.0	92.62	93.80
	MCC	0.08	0.12	0.10	0.13	0.08	0.12	0.09	0.10	0.08	0.10	0.17	0.19
Chebyshev	Target	63.82	65.95	36.17	27.65	64.89	63.82	42.55	32.97	62.76	63.82	41.48	36.17
	Outlier	51.06	56.28	86.54	88.96	55.56	64.48	81.82	86.43	51.38	61.59	77.40	83.53
	MCC	0.06	0.09	0.14	0.11	0.08	0.13	0.13	0.12	0.06	0.11	0.09	0.11
Manhattan	Target	68.08	67.02	35.10	31.91	64.89	64.89	47.87	38.29	68.08	67.02	51.06	39.36
	Outlier	48.43	58.40	89.20	91.74	54.21	60.58	65.54	76.81	50.44	55.39	69.49	76.34
	MCC	0.07	0.11	0.16	0.17	0.08	0.11	0.06	0.07	0.08	0.10	0.09	0.08

Mejores Modelos:	AMP	Target: EM-EUC	71.27	AB	Target: EM-EUC	67.02	Gram	Target: KMM-MHT	51.06
		Outlier: KME-CHB	55.56		Outlier: KME-CHB	64.48		Outlier: KMM-EUC	92.62
		MCC: KME-CHB	0.08		MCC: KME-CHB	0.13		MCC: KMM-EUC	0.17

Tabla 42. Resultados del modelo construido con descriptores starPep + iFeature #1 para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		Starpep + iFeature #1											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		128		131		79							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	75.66	67.26	12.48	11.39	75.58	66.83	26.41	23.75	78.60	70.83	26.55	25.76
	Outlier	36.57	54.21	75.01	87.45	44.16	55.17	61.65	79.55	40.83	51.90	43.96	70.09
	MCC	0.12	0.20	-0.14	-0.01	0.19	0.21	-0.12	0.03	0.19	0.21	-0.28	-0.04
Chebyshev	Target	71.53	69.62	9.05	7.95	74.96	69.99	21.20	20.44	75.46	70.15	14.27	13.95
	Outlier	42.25	48.79	60.88	79.39	41.02	55.96	60.95	81.00	38.93	52.57	77.78	88.72
	MCC	0.13	0.17	-0.31	-0.16	0.15	0.24	-0.18	0.017	0.14	0.21	-0.09	0.03
Manhattan	Target	73.13	66.53	36.88	29.93	73.49	70.28	33.99	30.89	76.65	71.29	24.89	22.79
	Outlier	38.09	50.05	37.92	64.85	46.39	53.06	45.96	71.03	38.13	47.83	42.13	64.57
	MCC	0.11	0.15	-0.24	-0.05	0.19	0.22	-0.19	0.02	0.15	0.18	-0.31	-0.13

Mejores Modelos:	AMP	Target: KMM-EUC	78.60	AB	Target: KMM-MHT	71.29	Gram	Target: EM-MHT	36.88
		Outlier: KME-MHT	46.39		Outlier: KME-CHB	55.96		Outlier: KMM-CHB	77.78
		MCC: KME-MHT	0.19		MCC: KME-CHB	0.24		MCC: KMM-CHB	-0.09

Tabla 43. Resultados del modelo construido con descriptores starPep + iFeature #1 para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		Starpep + iFeature #1											
Prueba:		Gram positivo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		128		131		118							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	75.66	67.26	30.15	26.07	75.58	66.83	27.62	24.04	78.60	70.83	20.86	20.33
	Outlier	36.57	54.21	65.42	78.33	44.16	55.17	62.85	76.31	40.83	51.90	88.72	90.22
	MCC	0.19	0.21	-0.09	0.00	0.19	0.21	0.12	0.14	0.13	0.17	0.05	0.09
Chebyshev	Target	71.53	69.62	30.65	25.95	74.96	69.99	38.16	35.31	75.46	70.15	43.86	39.07
	Outlier	42.25	48.79	74.25	82.12	41.02	55.96	71.34	78.29	38.93	52.57	66.43	74.98
	MCC	0.15	0.24	0.09	0.14	0.14	0.21	0.10	0.14	0.14	0.21	0.10	0.14
Manhattan	Target	73.13	66.53	28.07	23.97	73.49	70.28	28.09	25.76	76.65	71.29	39.75	36.66
	Outlier	38.09	50.05	83.88	87.24	46.39	53.06	48.29	67.32	38.13	47.83	51.59	64.85
	MCC	0.11	0.15	0.14	0.14	0.19	0.22	-0.22	-0.07	0.15	0.18	-0.08	0.01

Mejores Modelos:	AMP	Target: KMM-EUC	78.60	AB	Target: KMM-MHT	71.29	Gram	Target: KMM-CHB	43.86
		Outlier: KME-MHT	46.39		Outlier: KME-CHB	55.96		Outlier: KMM-EUC	88.72
		MCC: KME-MHT	0.19		MCC: EM-CHB	0.24		MCC: EM-MHT	0.14

Tabla 44. Resultados del modelo construido con descriptores starPep + iFeature #2 para el conjunto Gram negativo.

Identificador:		Starpep + iFeature #2											
Prueba:		Gram negativo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		30		27		21							
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	73.61	77.08	49.30	38.19	75.69	75.69	52.08	44.44	77.77	75.0	54.16	46.52
	Outlier	46.19	53.27	50.08	80.76	46.84	49.91	48.84	77.10	44.30	45.36	45.25	72.68
	MCC	0.10	0.16	-0.00	0.12	0.12	0.13	0.00	0.13	0.11	0.11	-0.00	0.11
Chebyshev	Target	75.69	73.61	33.33	25.69	76.38	73.61	42.36	33.33	77.08	73.61	40.27	34.02
	Outlier	45.19	57.40	80.82	92.38	46.66	49.55	80.82	90.02	44.48	46.84	69.97	85.13
	MCC	0.13	0.16	0.09	0.16	0.12	0.12	0.15	0.19	0.11	0.11	0.05	0.13
Manhattan	Target	71.52	77.08	33.33	29.86	77.77	68.75	34.02	29.16	79.86	76.38	47.91	38.88
	Outlier	44.30	52.21	50.02	81.12	47.37	49.73	77.05	89.26	44.77	47.19	65.13	84.07
	MCC	0.08	0.15	-0.08	0.07	0.13	0.09	0.06	0.15	0.13	0.12	0.07	0.16

Mejores Modelos:	AMP	Target: KMM-MHT	79.86	AB	Target: EM-MHT	77.08	Gram	Target: KMM-EUC	54.16
		Outlier: KME-MHT	47.37		Outlier: EM-CHB	57.40		Outlier: KME-CHB	88.82
		Precisión total: KME-MHT	0.13		Precisión total: EM-CHB	0.16		Precisión total: KME-CHB	0.15

Tabla 45. Resultados del modelo construido con descriptores starPep + iFeature #2 para el conjunto Gram positivo.

Identificador:		Starpep + iFeature #2											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		30		27		37							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	71.27	76.59	24.46	20.21	73.40	77.65	29.78	25.53	73.40	78.72	32.97	27.65
	Outlier	46.19	53.27	87.61	91.68	46.84	49.91	96.93	97.22	44.30	45.36	90.26	91.79
	MCC	0.07	0.13	0.08	0.09	0.09	0.12	0.28	0.26	0.07	0.10	0.16	0.15
Chebyshev	Target	71.27	69.14	38.29	30.85	69.14	75.53	40.42	30.85	69.14	79.78	47.87	35.10
	Outlier	45.19	57.40	87.07	91.68	46.66	49.55	77.69	84.24	44.48	46.84	75.45	82.18
	MCC	0.07	0.11	0.16	0.17	0.07	0.11	0.09	0.09	0.06	0.11	0.11	0.09
Manhattan	Target	72.34	74.46	19.14	13.82	74.46	72.34	26.59	22.34	70.21	76.59	34.04	27.65
	Outlier	44.30	52.21	88.96	92.62	47.37	49.73	91.74	93.92	44.77	47.19	94.21	95.69
	MCC	0.07	0.11	0.05	0.05	0.09	0.09	0.14	0.14	0.06	0.10	0.24	0.22

Mejores Modelos:	AMP	Target: KME-MHT	74.46	AB	Target: KMM-CHB	79.78	Gram	Target: KMM-CHB	47.87
		Outlier: KME-MHT	47.37		Outlier: EM-CHB	57.40		Outlier: KME-EUC	96.93
		MCC: KME-MHT	0.09		MCC: EM-EUC	0.13		MCC: KME-EUC	0.28

Tabla 46. Resultados del modelo construido con descriptores starPep + iFeature #2 para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		Starpep + iFeature #2											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		30		27		21							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	76.21	70.86	39.20	30.94	79.93	71.31	50.08	41.06	82.01	74.81	45.44	39.44
	Outlier	32.42	39.40	39.99	69.04	34.67	37.12	37.47	65.56	33.61	29.25	31.26	59.10
	MCC	0.09	0.10	-0.19	-0.00	0.15	0.08	-0.12	0.06	0.16	0.04	-0.22	-0.01
Chebyshev	Target	77.98	70.96	21.12	18.39	80.31	72.71	23.32	20.54	81.40	73.56	27.98	24.33
	Outlier	34.63	44.74	73.41	86.57	34.57	34.16	76.02	85.10	31.77	31.65	58.77	79.52
	MCC	0.13	0.15	-0.06	0.06	0.15	0.07	-0.00	0.07	0.14	0.05	-0.13	-0.02
Manhattan	Target	77.39	70.81	33.17	26.55	80.05	72.26	23.22	20.49	81.48	73.92	29.38	25.56
	Outlier	34.66	40.54	38.61	70.20	35.10	31.38	67.96	80.08	32.77	30.00	53.98	71.67
	MCC	0.12	0.11	-0.26	-0.03	0.15	0.03	-0.09	0.00	0.15	0.04	-0.16	-0.02

Mejores Modelos:	AMP	Target: KMM-EUC	82.01	AB	Target: KMM-EUC	74.81	Gram	Target: KME-EUC	50.08
		Outlier: KME-MHT	35.10		Outlier: EM-CHB	44.74		Outlier: KME-CHB	76.02
		MCC: KMM-EUC	0.16		MCC: EM-CHB	0.15		MCC: EM-CHB	0.06

Tabla 47. Resultados del modelo construido con descriptores starPep + iFeature #2 para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		Starpep & iFeature			
Prueba:		Gram positivo - Ext			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram positivo (375)	
Numero de descriptores:		30	27	37	
Conjuntos de prueba:		Positivo: Gram_Ext (5856)		Negativo: Ext_Neg (10771)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	76.21	70.86	19.65	17.16	79.93	71.31	10.55	9.39	82.01	74.81	19.77	18.42
	Outlier	32.42	39.41	75.95	83.06	34.67	37.12	94.72	95.78	33.61	29.25	85.30	87.54
	MCC	0.09	0.10	-0.05	0.00	0.15	0.08	0.09	0.10	0.16	0.04	0.06	0.08
Chebyshev	Target	77.98	70.96	31.55	26.22	80.31	72.71	33.93	29.57	81.40	73.56	37.27	32.66
	Outlier	34.63	44.74	79.51	86.48	34.57	34.16	69.99	78.08	31.77	31.65	67.17	74.75
	MCC	0.13	0.15	0.12	0.15	0.15	0.07	0.04	0.08	0.14	0.05	0.04	0.07
Manhattan	Target	77.39	70.81	21.63	18.86	80.05	72.26	21.02	19.62	81.48	73.92	21.12	19.72
	Outlier	34.66	40.54	77.94	96.05	35.10	31.38	88.30	90.14	32.77	30.00	89.98	91.88
	MCC	0.12	0.11	-0.00	0.06	0.15	0.03	0.12	0.13	0.15	0.04	0.15	0.16

Mejores Modelos:	AMP	Target: KMM-EUC	82.01	AB	Target: KMM-EUC	74.81	Gram	Target: KMM-MHT	37.27
		Outlier: KME-MHT	35.10		Outlier: EM-CHB	44.74		Outlier: KME-EUC	94.72
		MCC: KMM-EUC	0.16		MCC: EM-CHB	0.15		MCC: KMM-MHT	0.15

Tabla 48. Resultados del modelo construido con descriptores ProtDcal para el conjunto Gram negativo.

Identificador:		PrtoDcal			
Prueba:		Gram negativo			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram negativo (572)	
Numero de descriptores:		207	93	93	
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)		Negativo: AB_Neg (1695)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	89.58	85.41	59.72	52.77	93.05	81.94	61.80	57.63	91.66	81.25	63.88	56.94
	Outlier	57.58	79.46	88.79	90.08	57.81	80.47	91.09	91.74	34.10	77.64	90.56	91.15
	MCC	0.25	0.39	0.36	0.33	0.27	0.39	0.42	0.40	0.26	0.35	0.42	0.38
Chebyshev	Target	88.88	81.25	45.83	39.58	93.05	81.94	59.72	55.55	91.66	80.55	60.41	52.77
	Outlier	29.55	76.99	90.91	91.56	57.94	78.70	93.56	93.74	57.46	77.58	90.67	91.03
	MCC	0.25	0.36	0.30	0.27	0.27	0.37	0.46	0.43	0.26	0.35	0.40	0.35
Manhattan	Target	88.88	83.33	28.47	23.61	93.05	83.33	36.11	34.02	93.05	83.33	48.61	43.05
	Outlier	57.28	74.80	92.27	93.15	58.40	72.86	92.56	93.80	57.40	77.40	95.16	95.63
	MCC	0.24	0.34	0.19	0.16	0.27	0.32	0.26	0.26	0.27	0.36	0.42	0.39

Mejores Modelos:	AMP	Target: KMM-MHT	93.05	AB	Target: EM-EUC	85.41	Gram	Target: KMM-EUC	63.88
		Outlier: KME-MHT	58.40		Outlier: KME-EUC	80.47		Outlier: KMM-MHT	95.16
		MCC: KMM-MHT	0.27		MCC: EM-EUC	0.39		MCC: KME-CHB	0.46

Tabla 49. Resultados del modelo construido con descriptores ProtDcal para el conjunto Gram positivo.

Identificador:		PrtoDcal											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		207		93		93							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	85.10	58.51	31.91	20.21	87.23	69.14	31.91	25.53	82.97	71.27	36.17	27.65
	Outlier	57.58	79.46	97.99	98.34	57.81	80.47	98.17	98.23	57.34	77.64	96.63	96.93
	MCC	0.19	0.20	0.35	0.25	0.20	0.26	0.36	0.30	0.18	0.25	0.33	0.26
Chebyshev	Target	72.34	59.57	23.40	13.82	87.23	65.95	29.78	24.46	84.04	73.40	28.72	25.53
	Outlier	58.05	76.99	98.17	98.46	57.99	78.70	97.34	97.40	57.46	77.58	97.81	97.99
	MCC	0.13	0.18	0.28	0.18	0.20	0.23	0.30	0.25	0.18	0.26	0.31	0.29
Manhattan	Target	82.97	60.63	27.65	21.27	86.17	72.34	28.72	23.40	87.23	74.46	26.59	25.53
	Outlier	57.28	74.80	98.64	98.70	58.40	72.86	98.46	98.58	57.40	77.40	98.87	98.99
	MCC	0.18	0.17	0.35	0.29	0.20	0.22	0.35	0.30	0.20	0.26	0.36	0.36

Mejores Modelos:	AMP	Target: KMM-MHT	87.23	AB	Target: KMM-MHT	74.46	Gram	Target: KMM-EUC	36.17
		Outlier: KME-MHT	58.40		Outlier: KME-EUC	80.47		Outlier: KMM-MHT	98.87
		MCC: KME-MHT	0.20		MCC: KMM-MHT	0.26		MCC: KMM-MHT	0.36

Tabla 50. Resultados del modelo construido con descriptores ProtDcal para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		PrtoDcal											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		207		93		93							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	86.01	77.30	38.09	36.27	91.13	79.43	45.76	45.30	90.64	79.11	48.05	47.30
	Outlier	48.78	66.76	79.91	81.75	48.57	67.73	83.23	84.25	48.34	63.96	81.18	82.31
	MCC	0.34	0.42	0.19	0.19	0.39	0.45	0.31	0.32	0.39	0.41	0.30	0.31
Chebyshev	Target	84.87	77.85	37.90	36.01	90.91	79.90	37.73	37.53	90.26	80.58	43.86	43.34
	Outlier	48.97	62.51	83.39	84.38	48.78	66.47	86.87	87.36	48.33	62.81	83.30	84.25
	MCC	0.33	0.38	0.23	0.23	0.39	0.44	0.28	0.28	0.38	0.41	0.29	0.30
Manhattan	Target	85.72	78.82	12.05	11.18	90.60	81.26	22.69	22.59	90.40	80.84	21.37	20.38
	Outlier	48.76	61.50	85.29	86.64	48.70	64.64	84.69	86.81	48.37	68.22	89.65	90.87
	MCC	0.34	0.38	-0.03	-0.03	0.39	0.43	0.09	0.12	0.38	0.46	0.15	0.15

Mejores Modelos:	AMP	Target: KME-EUC	91.13	AB	Target: KME-MHT	81.26	Gram	Target: KMM-EUC	48.05
		Outlier: EM-CHB	48.97		Outlier: KMM-MHT	68.22		Outlier: KMM-MHT	89.65
		MCC: KME-EUC	0.39		MCC: KMM-MHT	0.46		MCC: KME-EUC	0.31

Tabla 51. Resultados del modelo construido con descriptores ProtDcal para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		PrtoDcal			
Prueba:		Gram positivo - Ext			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram positivo (375)	
Numero de descriptores:		207	93	93	
Conjuntos de prueba:		Positivo: Gram_Ext (5856)		Negativo: Ext_Neg (10771)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	86.01	77.30	16.51	14.99	91.13	79.43	22.67	22.38	90.64	79.11	29.50	29.13
	Outlier	48.78	66.77	95.14	95.73	48.57	67.73	96.21	96.41	48.34	63.96	92.88	93.30
	MCC	0.34	0.42	0.19	0.18	0.39	0.45	0.29	0.29	0.39	0.41	0.29	0.30
Chebyshev	Target	84.87	77.85	14.15	13.11	90.91	79.90	21.70	21.51	90.26	80.58	22.72	22.40
	Outlier	48.97	62.51	95.58	95.82	48.78	66.47	96.17	96.31	48.33	62.81	97.32	97.56
	MCC	0.33	0.38	0.17	0.16	0.39	0.44	0.28	0.28	0.38	0.41	0.32	0.32
Manhattan	Target	85.72	78.82	12.92	12.31	90.60	81.26	23.29	23.10	90.40	80.84	25.87	25.70
	Outlier	48.76	61.50	97.00	97.11	48.70	64.64	98.26	98.30	48.37	68.22	98.99	99.01
	MCC	0.34	0.38	0.19	0.18	0.39	0.43	0.35	0.35	0.38	0.46	0.40	0.39

Mejores Modelos:	AMP	Target: KME-EUC	91.13	AB	Target: KME-MHT	81.26	Gram	Target: KMM-EUC	29.50
		Outlier: EM-CHB	48.97		Outlier: KMM-MHT	68.22		Outlier: KMM-MHT	98.99
		MCC: KME-EUC	0.39		MCC: KMM-MHT	0.46		MCC: KMM-MHT	0.40

Tabla 52. Resultados del modelo construido con la codificación BERT-ESM #1 para el conjunto Gram negativo.

Identificador:		BERT ESM #1			
Prueba:		Gram negativo			
Conjuntos de entrenamiento:		AMP (8803)	AB (4844)	Gram negativo (572)	
Numero de descriptores:		7	6	6	
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)		Negativo: AB_Neg (1695)	

Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	61.11	64.58	41.66	20.83	65.27	61.11	32.63	14.58	70.83	70.83	39.58	20.13
	Outlier	33.45	49.43	75.33	84.12	33.21	48.61	75.98	84.48	26.43	39.76	81.47	85.60
	MCC	-0.03	0.07	0.10	0.03	-0.00	0.05	0.05	-0.00	-0.01	0.05	0.14	0.04
Chebyshev	Target	59.02	55.55	34.72	13.19	59.72	52.77	38.19	15.27	68.75	50.0	42.36	15.97
	Outlier	34.57	54.86	76.57	86.60	31.91	55.45	75.45	84.48	27.43	57.75	68.96	80.70
	MCC	-0.03	0.05	0.07	-0.00	-0.04	0.04	0.08	-0.00	-0.02	0.04	0.06	-0.02
Manhattan	Target	59.72	56.25	36.80	18.05	64.58	55.55	36.11	15.27	71.52	62.5	37.5	19.44
	Outlier	34.63	54.10	77.81	88.08	35.63	51.09	82.30	88.67	26.19	44.54	69.55	79.11
	MCC	-0.03	0.05	0.09	0.05	0.00	0.03	0.12	0.03	-0.01	0.03	0.04	-0.00

Mejores Modelos:	AMP	Target: KMM-MHT	71.52	AB	Target: KMM-EUC	70.83	Gram	Target: KMM-CHB	42.36
		Outlier: MHT-KME	35.63		Outlier: KME-EUC	57.75		Outlier: KME-MHT	82.30
		MCC: KME-MHT	0.00		MCC: EM-EUC	0.07		MCC: KMM-EUC	0.014

Tabla 53. Resultados del modelo construido con la codificación BERT-ESM #1 para el conjunto Gram positivo.

Identificador:		BERT ESM #1											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		7		6		7							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	74.46	62.76	37.23	26.59	65.95	65.95	39.36	30.85	76.59	68.08	41.48	30.85
	Outlier	33.45	49.43	53.15	75.81	33.21	48.61	70.67	81.23	26.43	39.76	64.30	74.51
	MCC	0.03	0.05	-0.04	0.01	-0.00	0.06	0.04	0.06	0.01	0.03	0.02	0.02
Chebyshev	Target	71.27	61.70	23.40	18.05	65.95	63.82	24.46	18.08	75.53	56.38	32.97	25.51
	Outlier	34.57	54.86	72.80	85.19	31.91	55.45	81.00	89.20	27.43	57.75	71.38	83.95
	MCC	-0.00	0.06	0.04	0.06	0.01	0.03	0.02	0.02	0.01	0.06	0.021	0.05
Manhattan	Target	68.08	59.57	29.78	19.14	67.02	61.70	23.40	19.14	73.40	64.89	24.46	18.08
	Outlier	34.63	54.10	72.09	83.83	35.63	51.09	83.83	90.26	26.19	44.54	78.93	85.72
	MCC	0.012	0.06	0.00	0.01	0.012	0.05	0.04	0.06	-0.00	0.04	0.01	0.024

Mejores Modelos:	AMP	Target: KMM-EUC	76.59	AB	Target: KMM-EUC	68.08	Gram	Target: KMM-EUC	41.48
		Outlier: KME-MHT	35.63		Outlier: KMM-CHB	57.75		Outlier: KMM-MHT	78.93
		MCC: EM-EUC	0.03		MCC: KME-EUC	0.06		MCC: KMM-MHT	0.04

Tabla 54. Resultados del modelo construido con la codificación BERT-ESM #1 para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		BERT ESM #1											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		7		6		6							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	67.72	59.27	27.11	16.32	69.60	60.19	26.24	16.18	74.41	66.90	28.92	19.55
	Outlier	37.48	55.77	79.98	88.03	37.04	54.54	80.47	88.22	29.41	43.79	83.41	88.34
	MCC	0.05	0.14	0.08	0.06	0.06	0.14	0.07	0.06	0.04	0.10	0.14	0.10
Chebyshev	Target	67.94	52.25	24.23	13.49	69.15	54.14	20.26	12.58	73.39	45.08	24.77	13.13
	Outlier	37.99	58.75	81.06	89.57	35.64	58.92	79.30	87.61	31.03	62.33	74.70	85.30
	MCC	0.05	0.10	0.06	0.04	0.04	0.12	-0.00	0.00	0.04	0.07	-0.00	-0.02
Manhattan	Target	66.46	54.32	27.23	15.02	67.93	54.83	21.36	12.43	72.72	63.13	27.68	18.83
	Outlier	39.60	56.95	81.93	90.68	40.46	55.05	85.84	91.43	30.22	49.66	73.62	83.05
	MCC	0.05	0.10	0.10	0.08	0.08	0.09	0.09	0.06	0.03	0.12	0.01	0.02

Mejores Modelos:	AMP	Target: KMM-EUC	74.41	AB	Target: KMM-EUC	66.90	Gram	Target: KMM-EUC	28.92
		Outlier: KME-MHT	40.46		Outlier: KMM-CHB	62.33		Outlier: KMM-MHT	85.84
		MCC: KME-MHT	0.08		MCC: KMM-MHT	0.14		MCC: KME-EUC	0.14

Tabla 55. Resultados del modelo construido con la codificación BERT-ESM #1 para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		BERT ESM #1											
Prueba:		Gram positivo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		7		6		7							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		**EM**				**K means Euclidean**				**K means Manhattan**			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	67.72	59.27	44.46	27.78	69.60	60.19	32.51	22.04	74.41	66.90	35.33	26.57
	Outlier	37.48	55.77	51.47	75.97	37.04	54.54	70.98	83.12	29.41	43.79	64.21	74.44
	MCC	0.05	0.14	-0.03	0.04	0.06	0.14	0.03	0.06	0.04	0.10	-0.00	0.01
Chebyshev	Target	67.94	52.25	29.64	18.59	69.15	54.14	21.65	14.44	73.39	45.08	27.71	15.23
	Outlier	37.99	58.75	74.83	86.16	35.64	58.92	80.73	89.23	31.03	62.33	71.89	85.08
	MCC	0.05	0.10	0.04	0.06	0.04	0.12	0.02	0.05	0.04	0.07	-0.00	0.00
Manhattan	Target	66.46	54.32	29.83	18.10	67.93	54.83	15.06	10.09	72.72	63.13	23.66	17.12
	Outlier	39.60	56.95	73.80	85.73	40.46	55.05	83.49	89.56	30.22	49.66	79.98	87.33
	MCC	0.05	0.10	0.03	0.05	0.08	0.09	-0.01	-0.00	0.03	0.12	0.04	0.06

Mejores Modelos:	AMP	Target: KMM-EUC	74.41	AB	Target: KMM-EUC	66.90	Gram	Target: EM-EUC	44.46
		Outlier: KME-MHT	40.46		Outlier: KMM-MHT	62.33		Outlier: KMM-CHN	79.98
		MCC: KME-EUC	0.08		MCC: KMM-MHT	0.14		MCC: KMM-MHT	0.04

Tabla 56. Resultados del modelo construido con la codificación BERT-ESM #2 para el conjunto Gram negativo.

Identificador:		Bert ESM #2											
Prueba:		Gram negativo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		7		6		6							
Conjuntos de prueba:		Positivo: Gram_negativo_test (144)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		*EM*				*K means Euclidean*				*K means Manhattan*			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	61.11	64.58	41.66	20.83	65.27	61.11	32.63	14.58	70.83	70.83	39.58	20.13
	Outlier	33.45	49.43	75.33	84.12	33.21	48.61	75.98	84.48	26.43	39.76	81.47	85.60
	MCC	-0.03	0.07	0.10	0.03	-0.00	0.05	0.05	-0.00	-0.01	0.05	0.14	0.04
Chebyshev	Target	59.02	55.55	34.72	13.19	59.72	52.77	38.19	15.27	68.75	50.0	42.36	15.97
	Outlier	34.57	54.86	76.57	86.60	31.91	55.45	75.45	84.48	27.43	57.75	68.96	80.70
	MCC	-0.03	0.05	0.07	-0.00	-0.04	0.04	0.08	-0.00	-0.02	0.04	0.06	-0.02
Manhattan	Target	59.72	56.25	36.80	18.05	64.58	55.55	36.11	15.27	71.52	62.5	37.5	19.44
	Outlier	34.63	54.10	77.81	88.08	35.63	51.09	82.30	88.67	26.19	44.54	69.55	79.11
	MCC	-0.03	0.05	0.09	0.05	0.00	0.03	0.12	0.03	-0.01	0.03	0.04	-0.00

Mejores Modelos:	AMP	Sensibilidad: KMM-MHT	71.52	AB	Sensibilidad: KMM-EUC	70.83	Gram	Sensibilidad: KMM-CHB	42.36
		Especificidad: MHT-KME	35.63		Especificidad: KME-EUC	57.75		Especificidad: KME-MHT	82.30
		MCC: KME-MHT	0.00		MCC: EM-EUC	0.07		MCC: KMM-EUC	0.014

Tabla 57. Resultados del modelo construido con la codificación BERT-ESM #2 para el conjunto Gram positivo.

Identificador:		Bert ESM #2											
Prueba:		Gram positivo											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		1280		1280		1280							
Conjuntos de prueba:		Positivo: Gram_positivo_test (94)				Negativo: AB_Neg (1695)							
Tipo de distancia / Clúster:		*EM*				*K means Euclidean*				*K means Manhattan*			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	58.51	57.44	26.59	18.08	58.51	58.51	29.78	23.40	58.51	57.44	34.04	24.46
	Outlier	43.89	44.24	64.07	67.49	38.76	42.47	57.93	63.30	38.17	32.97	49.26	57.40
	MCC	0.01	0.00	-0.04	-0.06	-0.01	0.00	-0.05	-0.06	-0.01	-0.04	-0.07	-0.08
Chebyshev	Target	58.51	54.25	28.72	23.40	60.63	56.38	42.55	29.78	61.70	59.47	42.55	27.65
	Outlier	40.11	49.26	62.12	65.78	37.46	40.82	61.23	65.36	35.81	42.12	57.69	62.94
	MCC	-0.00	0.01	-0.04	-0.05	-0.00	-0.01	-0.01	-0.02	-0.01	0.00	0.00	-0.04
Manhattan	Target	74.46	67.02	21.27	18.08	64.89	58.51	24.46	20.21	63.82	59.57	20.21	12.76
	Outlier	22.24	31.85	92.44	92.44	33.74	38.34	80.29	80.35	37.40	35.33	97.52	97.58
	MCC	-0.01	-0.00	0.11	0.08	-0.00	-0.01	0.02	0.00	0.00	-0.02	0.21	0.13

Mejores Modelos:	AMP	Sensibilidad: EM-MHT	74.46	AB	Sensibilidad: EM-EUC	67.02	Gram	Sensibilidad: KME-CHB	42.55
		Especificidad: EM-EUC	43.89		Especificidad: EM-CHB	49.26		Especificidad: KMM-MHT	97.52
		MCC: EM-EUC	0.01		MCC: EM-CHB	0.01		MCC: KMM-MHT	0.21

Tabla 58. Resultados del modelo construido con la codificación BERT-ESM #2 para los conjuntos externos utilizando la etapa Gram negativo.

Identificador:		BERT ESM #2											
Prueba:		Gram negativo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram negativo (572)							
Numero de descriptores:		1280		1280		1280							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		*EM*				*K means Euclidean*				*K means Manhattan*			
		AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer	AMP	AB	Gram-	Jer
Euclidean	Target	67.17	59.10	1.43	1.40	64.32	60.92	3.31	2.20	64.80	61.15	3.60	2.61
	Outlier	44.45	45.76	99.62	99.62	41.32	43.89	99.87	99.93	40.87	36.70	99.85	99.92
	MCC	0.11	0.04	0.05	0.05	0.05	0.04	0.13	0.11	0.05	-0.02	0.14	0.12
Chebyshev	Target	65.94	54.09	19.87	18.11	66.44	55.02	21.84	19.77	65.96	55.43	24.14	20.69
	Outlier	41.22	50.10	72.06	73.59	39.64	43.36	66.25	69.86	36.83	44.46	66.58	71.24
	MCC	0.07	0.04	-0.08	-0.09	0.06	-0.01	-0.12	-0.11	0.02	-0.00	-0.09	-0.08
Manhattan	Target	73.90	62.00	15.88	15.65	73.85	62.29	13.88	13.66	73.75	62.97	16.23	15.43
	Outlier	26.99	34.88	80.53	80.62	38.42	41.75	85.09	85.11	41.18	39.42	83.42	83.46
	MCC	0.00	-0.03	-0.04	-0.04	0.12	0.03	-0.01	-0.01	0.014	0.02	-0.00	-0.01

Mejores Modelos:	AMP	Sensibilidad: EM-MHT	73.90	AB	Sensibilidad: KMM-MHT	62.97	Gram	Sensibilidad: KMM-EUC	28.92
		Especificidad: EM-EUC	44.45		Especificidad: EM-CHB	50.10		Especificidad: KMM-MHT	85.84
		MCC: KME-MHT	0.12		MCC: KMM-EUC	0.14		MCC: KME-EUC	0.14

Tabla 59. Resultados del modelo construido con la codificación BERT-ESM #2 para los conjuntos externos utilizando la etapa Gram positivo.

Identificador:		BERT ESM #2											
Prueba:		Gram positivo - Ext											
Conjuntos de entrenamiento:		AMP (8803)		AB (4844)		Gram positivo (375)							
Numero de descriptores:		1280		1280		1280							
Conjuntos de prueba:		Positivo: Gram_Ext (5856)				Negativo: Ext_Neg (10771)							
Tipo de distancia / Clúster:		EM				K means Euclidean				K means Manhattan			
		AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer	AMP	AB	Gram+	Jer
Euclidean	Target	67.17	59.10	18.83	17.16	64.32	60.92	22.60	20.33	64.80	61.15	26.05	21.70
	Outlier	44.45	45.76	61.19	64.21	41.32	43.89	56.16	63.39	40.87	36.70	47.43	56.36
	MCC	0.11	0.04	-0.20	-0.19	0.05	0.04	-0.21	-0.16	0.05	-0.02	-0.25	-0.21
Chebyshev	Target	65.94	54.09	21.07	18.34	66.44	55.02	24.70	20.67	65.96	55.43	24.67	19.72
	Outlier	41.27	50.16	61.97	64.64	39.64	43.36	63.48	66.66	36.83	44.46	58.10	62.20
	MCC	0.07	0.04	-0.17	-0.17	0.06	-0.01	-0.12	-0.13	0.02	-0.00	-0.17	-0.18
Manhattan	Target	73.90	62.04	5.05	4.78	73.85	62.29	11.78	11.61	73.75	62.97	5.58	5.22
	Outlier	26.99	34.88	91.86	91.90	38.42	41.75	80.97	81.24	41.18	39.42	97.40	97.47
	MCC	0.00	-0.03	-0.05	-0.06	0.12	0.03	-0.09	-0.09	0.14	0.02	0.07	0.07

Mejores Modelos:	AMP	Sensibilidad: EM-MHT	73.90	AB	Sensibilidad: KMM-EUC	62.97	Gram	Sensibilidad: KMM-EUC	26.05
		Especificidad: EM-EUC	44.45		Especificidad: EM-CHB	50.16		Especificidad: KMM-MHT	97.40
		MCC: KMM-MHT	0.14		MCC: EM-CHB	0.04		MCC: KMM-MHT	0.07