MODELO DE PREDICCIÓN DE PÉPTIDOS ANTIMICROBIANOS BASADO EN TÉCNICAS DE APRENDIZAJE COMPUTACIONAL Y ESTADÍSTICO

Francy Liliana Camacho Urrea

Ingeniera de Sistemas

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga
2016

MODELO DE PREDICCIÓN DE PÉPTIDOS ANTIMICROBIANOS BASADO EN TÉCNICAS DE APRENDIZAJE COMPUTACIONAL Y ESTADÍSTICO

Francy Liliana Camacho Urrea
Ingeniera de Sistemas

Trabajo de grado presentado para optar por el título de
Magíster en Ingeniería de Sistemas e Informática

Directores del Proyecto:
Raúl Ramos Pollán, *PhD*
Rodrigo Gonzalo Torres Sáez, *PhD*

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga
2016

A mis padres

Agradecimientos

A mis directores Rodrigo Torres y Raúl Ramos por orientarme a lo largo del proyecto y por apoyarme en este tiempo.

A la profesora Claudia Cristina Ortiz por ofrecerme el espacio y los recursos para desarrollar el proyecto.

A Paola Rondón, de la E3T por su asesoría en los software para el cálculo de los descriptores.

A Marlon Cáceres, Alba López , Yuly Prada, Geovanni Santamaría Y Jennifer Cruz del grupo GIBIM, por proveerme las secuencias de péptidos.

A mis amigos del GIBIM y SC3 por hacer muy agradable este tiempo.

A Darío por acompañarme, escucharme y aconsejarme en tantos momentos.

A todos aquellos que no nombré, pero que contribuyeron en mi proceso de formación.

Mil gracias a todos.

Glosario

- **Aprendizaje de características:** es un conjunto de técnicas que transforman los datos de entrada en una nueva representación, para luego ser usado de forma más efectiva en tareas de aprendizaje automático.
- **Catiónico:** Hace referencia a la carga positiva de un átomo o molécula.
- **Máquinas de Soporte Vectorial:** Son algoritmos de optimización asociados al aprendizaje supervisado, en el cual, a partir de un conjunto de entrenamiento crean modelos matemáticos capaces de clasificar un nuevo elemento sin conocer de antemano la clase a la que pertenece.
- **Péptidos antibacterianos:** Son proteínas de origen natural que tienen propiedades antibióticas y que han sido fabricados por la naturaleza para actuar como medio de defensa en contra de enfermedades producidas por bacterias.
- **Resistencia bacteriana:** Es la capacidad de una bacteria para resistir los efectos de un medicamento.
- **Péptido:** Son un tipo de moléculas formadas por la unión de varios aminoácidos mediante enlaces peptídicos.
- **Polipéptido:** es el nombre utilizado para designar un péptido de tamaño suficientemente grande.
- **Selección de características:** es un conjunto de técnicas usadas para reducir el número de características de entrada a un tamaño apropiado para luego ser procesadas y analizadas.
- **Sparse autoencoder:** es una clase especial de red neuronal, usada de forma no supervisada.
- **Stacked autoencoder:** es una red neuronal con dos o más capas ocultas de autoencoders, cuyo aprendizaje se realiza de forma no supervisada.

Índice general

Índice de figuras

Lista de Tablas

Lista de Anexos

Resumen

TITULO: Modelo de predicción de péptidos antimicrobianos basado en técnicas de aprendizaje computacional y estadístico [1]

AUTOR: Francy Liliana Camacho Urrea. [2]

PALABRAS CLAVE: Diseño de medicamentos, regresión de vectores de soporte, péptidos antimicrobianos, autoencoders, stacked autoencoder

En los últimos años, el reconocimiento de patrones se ha aplicado en diversas áreas para resolver múltiples problemas. Una de estas áreas es el diseño *in silico* de medicamentos, donde ha sido ampliamente utilizados en el análisis de proteínas. Por ejemplo, para predecir la actividad antimicrobiana presente en péptidos (proteínas cortas), los cuales se están utilizando como alternativas a los medicamentos tradicionales. En este trabajo, se propone utilizar herramientas como la Regresión de Vectores de Soporte (SVR, por sus siglas en inglés) junto con el modelo denominado Relación Cuantitativa entre Estructura-Actividad (QSAR, por sus siglas en inglés), para realizar el reconocimiento de patrones y crear algoritmos que permitan predecir la actividad antimicrobiana en péptidos.

Para tal fin, se propuso una metodología que integra aprendizaje y selección de características junto con métricas de rendimiento aplicado a dos conjuntos de datos (uno con alta identidad y otro con baja identidad). Ésta incluyó stacked autoencoders, algoritmos genéticos, curvas de aprendizaje, prueba de permutación, etc. A través de esta metodología, los predictores obtenidos con el primer conjunto de datos son estadísticamente estables y mostraron rendimientos competitivos con respecto a la literatura. Sin embargo, con el segundo grupo los resultados mostrados fueron bajos en cada métrica. Por lo anterior, nuestro aporte se centró en el aprendizaje de características, identificación de problemas de sesgo o alta varianza y confiabilidad estadística de los modelos predictivos, así como la definición de los requerimientos del problema para que esta estrategia funcione correctamente.

[1] Trabajo de grado

[2] Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Raúl Ramos. Codirector: Rodrigo Torres.

Abstract

TITLE: Prediction model of antimicrobial peptides based on machine learning and statistical learning[3]

AUTHOR: Francy Liliana Camacho[4]

KEY WORDS: Design of drugs, support vector regression, antimicrobial peptides, autoencoders, stacked autoencoder.

In recent years, the pattern recognition has been applied in many areas to solve diverse problems. One of those areas is the *in silico* drug design, which has been widely used in the protein analysis. For example, to predict the antimicrobial activity in peptides (small proteins), which are being used as alternatives to traditional medicines. In this paper, we propose to use tools such as the Support Vector Regression (SVR) and Quantitative Structure-Activity Relationship (QSAR) model, for recognition patterns and create algorithms to predict the antimicrobial activity in peptides.

For this purpose, we proposed a methodology that integrates feature learning and selection methods and thorough use of performance metrics applied on two datasets (first one with high identity and the other with low identity). This includes stacked autoencoders, genetic algorithms, learning curves, permutation tests, etc. Through this methodology, the predictors we obtain are statistically stable and they have competitive performance with respect to literature. However, with the second group the results shown were low in each metric. Therefore, our contribution focused on feature learning, identifying problems of bias or high variance and statistical stability, and the definition of the requerements of the problem for this strategy to work properly.

[3]Research work

[4]Faculty of Physical-Mechanical Engineerings. Systems engineering and informatics department. Advisor: Raúl Ramos. Co-advisor: Rodrigo Torres

Introducción

En las últimas décadas, a pesar de los avances en el tratamiento de enfermedades, especialmente aquellas causadas por patógenos, existe un creciente aumento de la resistencia de los microorganismos a múltiples medicamentos, lo que se ha convertido en un problema de salud pública mundial, según la Organización Mundial de la Salud Gilbert et al. (2010). Por esta razón, la búsqueda de nuevos fármacos que combatan la resistencia de los microorganismos patógenos, se ha convertido en una prioridad de salud pública. Entre éstos, los péptidos antimicrobianos se consideran prometedores, debido principalmente, a su baja probabilidad de generar resistencia bacteriana, rápida acción y amplio espectro de actividad Marr et al. (2006). Sin embargo, actualmente se trabaja en la búsqueda de nuevos péptidos artificiales más potentes que los actuales, aunque se ve limitado por el gran cantidad de arreglos posibles que se pueden generar, si se tienen en cuenta los 20 aminoácidos naturales Fjell et al. (2012).

Para abordar la solución a este problema, se emplean técnicas computacionales como Relaciones Cuantitativas Estructura-Actividad (QSAR, por sus siglas en inglés), cuya premisa se basa en que un péptido puede ser descrito a través de sus propiedades físico-químicas, llamadas descriptores y éstas se correlacionan con su actividad (que puede ser una variable continua, como la Concentración Mínima Inhibitoria (CMI) o una variable categórica como el tipo de actividad por ejemplo, antifúngica, antiviral, etc.), a través de la construcción de una función matemática Cherkasov (2005),Dudek et al. (2006), Taboureau (2010).

A partir de lo anterior, en esta propuesta se busca crear un modelo para predecir la actividad antimicrobiana de péptidos usando técnicas de aprendizaje computacional y estadístico, para tratar de reducir el número de secuencias que podrían ser candidatas a sintetizar. Para describir el trabajo desarrollado, este documento se articula en cuatro capítulos principales en los que se aborda, en la primera sección los antecedentes y contextualización del tema, en el segun-

do capítulo se propone una metodología que integra aprendizaje y selección de características (basado en stacked autoencoders y algoritmos genéticos, respectivamente) junto con regresión de vectores de soporte, para obtener mejores correlaciones entre los descriptores y la actividad, así mismo, debido a que los conjuntos de datos empleados son pequeños (apenas unas cientos de muestras), se planteó un esquema de validación para que los resultados de los modelos creados, sean estadísticamente estables. En la sección 3, se prueba esta metodología con el grupo de secuencias obtenidas en el laboratorio GIBIM y finalmente se presentan las conclusiones generales del trabajo realizado.

1. Marco de referencia

1.1. La necesidad de nuevos antibióticos

En los últimos años, se ha visto la necesidad de desarrollar nuevos antibióticos debido a la aparición y diseminación de cepas resistentes. Dicha resistencia es producto de diversas variables biológicas (las bacterias tienen características y habilidades genéticas que permiten una rápida evolución hacia la resistencia), sociales (abuso de antibióticos) y farmacológicas (los pacientes suspenden el tratamiento de forma prematura cuando presentan mejoría) en todo el mundo. Sin embargo, se agravan en los países en vías de desarrollo Amábile-Cuevas (2010)

Así mismo, se ha demostrado que hay pocos medicamentos que ofrecen mejores beneficios que los actuales y muchos menos que se enfoquen en el tratamiento de enfermedades infecciosas como las producidas por la *Staphylococcus aureus*, *Pseudomonas aeruginosa*, entre otras. En vista de ello se han propuesto estrategias para tratar estos problemas como la iniciativa "10x20", que consiste en desarrollar 10 nuevos fármacos antibacterianos para el año 2020 Gilbert et al. (2010).

1.2. Alternativas a los antibióticos

Debido a estos problemas, se han estudiado diversas alternativas a los antibióticos actuales, entre las que se destacan los péptidos antibacterianos, que son proteínas cortas con tamaños entre 12 a 50 aminoácidos y hacen parte del sistema inmune de muchos seres vivos. La mayor parte de ellos son anfipáticos, tienen carga neta positiva y contienen alrededor del 50 % de aminoácidos hidrofóbicos, características que les permiten interactuar con la membrana negativa de las bacterias.

Se caracterizan principalmente a su baja probabilidad de generar resistencia bacteriana, sin embargo el diseño y síntesis de nuevos péptidos se ve limitado, entre otras razones, al gran número de arreglos posibles que se pueden generar, por ejemplo, teniendo en cuenta los 20 aminoácidos naturales, si se quiere elaborar un péptido cuyo tamaño tenga 6 aminoácidos, el número de combinaciones posibles se traduce a varios órdenes de magnitud (si $n = 6, 20^6 = 6{,}4 \times 10^7$) y no es factible probarlas todas en el laboratorio Fjell et al. (2012). Por ésta razón se ha abordado enfoques computaciones como el modelo Relaciones Cuantitativas entre Estructura-Actividad (QSAR, por sus siglas en inglés), ampliamente usado en la predicción de péptidos antimicrobianos.

1.3. Relación Cuantitativa entre Estructura-Actividad (QSAR)

QSAR (por sus siglas en inglés) es un método computacional empleado ampliamente en el diseño de nuevas moléculas usadas como medicamentos, sin embargo, recientemente se ha aplicado en péptidos antimicrobianos Jenssen (2011). QSAR se basa en la idea de que un péptido puede ser descrito a través de sus propiedades físico-químicas y estructurales, denominadas descriptores moleculares, y que éstas, a su vez, pueden ser correlacionadas con la actividad antimicrobiana del péptido a través de una función matemática Taboureau (2010). La finalidad de enlazar las propiedades físico-químicas del péptido con su actividad biológica es un intento por entender mejor el mecanismo de acción de éstos y permitir el diseño de nuevos péptidos más potentes Jenssen (2011). Sin embargo, el éxito de QSAR depende de varios factores:

1.3.1. Colección de datos biológicos

En la recopilación de datos biológicos se espera que los datos provengan de la misma fuente, que hayan sido evaluados con el mismo protocolo de experimentación, laboratorio, material y personal Scior et al. (2009). Por ende, los datos que se emplean en diversos estudios QSAR son pequeños, con apenas unos cientos de muestras, que puede generar dificultades en algunos modelos Zhou et al. (2010), Shu et al. (n.d.), Borkar et al. (2013).

1.3.2. Caracterización de las secuencias: Descriptores

Los descriptores moleculares son el resultado final de un procedimiento lógico y matemático que transforma la información química presente en una molécula, en una medida cuantitativa Todeschini and Consonni (2000), algunos de los cuales se obtienen a partir de un proceso experimental y otros se calculan con base en la secuencia o su estructura. No obstante, existen más de 1000 descriptores obtenidos a partir de la estructura primaria, secundaria y terciaria.

Así mismo, existe otra forma de clasificar el tipo de descriptores, como los 0D que se calculan a partir de la fórmula molecular (número de átomos, carbonos, hidrógenos, etc), 1D que se extraen a partir de la secuencia (número de anillos, H-bond acceptor or donor, etc), los 2D que tienen en cuenta aspectos topológicos de la estructura (Balaban, Randic, Wiener), los 3D se calculan de la geometría molecular y propiedades de la superficie (GETAWAY, autocorrelación, tamaño, superficie, volumen, etc) y los 4D que tiene en cuenta las energías de interacción entre coordenadas 3D analizadas en una malla Jenssen (2011).

1.3.3 Preprocesamiento de características

El pre-procesamiento es un paso previo al tratamiento de los datos con las máquinas de aprendizaje y consiste en mejorar la calidad de los datos usando diversas estrategias. Esto debido a que, normalmente, los datos que se emplean para procesamiento traen consigo ruido, valores faltantes o disparidad en los rangos, lo que puede afectar de forma significativa los resultados, por ende, se emplean varias técnicas para tratar estos problemas. Por ejemplo, en este trabajo se realizó un preprocesamiento que incluyó la remoción de aquellos descriptores donde el valor de la característica fuera el mismo en todos los péptidos (la desviación del descriptor igual a 0) y luego se estandarizó (μ=0 y δ=1) cada uno éstos, debido a las amplias variaciones Zhou et al. (2010).

1.3.4 Selección de características

Como se ha descrito previamente, el número de descriptores que se pueden obtener a partir del péptido es amplio y se requiere seleccionar aquellos que son relevantes y que describen mejor la dependencia entre descriptores y la actividad, puesto que en el modelo, algunos pueden ser considerados como ruido, otros redundantes y estar correlacionados. Adicionalmente, con la selección se

obtienen otras ventajas como facilidad de visualizar y entender los datos, reducción del tiempo de entrenamiento y en algunos casos aumento del rendimiento.

Uno de los métodos empleados en este trabajo fue los algoritmos genéticos, que son algoritmos de optimización inspirados en el principio de la evolución natural. Es decir, a partir de una población de individuos (posibles soluciones), cada miembro denominado cromosoma, representa un grupo de características seleccionadas que generalmente se expresa como un vector binario (1 si se va a tener en cuenta la característica, 0 de lo contrario). A partir de ello se evalúa en una función objetivo que determina el valor de aptitud del individuo. Durante el curso de la evolución, los cromosomas están sujetos a la selección, cruce y mutación. En la selección, se escogen los individuos que pasarán a la siguiente generación, dando una mayor probabilidad a aquellos con el valor de aptitud más alto. En el cruce, se comparte la información genética de la población entre si, de acuerdo a si dos individuos superan o no una probabilidad. La mutación corresponde a la alteración de la información genética del individuo (una posición en el cromosoma).

1.3.5 Mapeo de descriptores y la actividad- Generación del modelo

El último paso en el modelo consiste en relacionar los descriptores (previamente preprocesados y seleccionados) con la actividad presente en el péptido. En el caso de los modelos de regresión, la variable dependiente (actividad, entendida como el CMI) es modelado en función de variables independientes (descriptores).

Uno de los modelos más empleados son las máquinas de soporte vectorial (SVM, por sus siglas en inglés), que son algoritmos de optimización asociados al aprendizaje supervisado, en el cual, a partir de un conjunto de entrenamiento crean modelos matemáticos capaces de clasificar un nuevo elemento sin conocer de antemano la clase a la que pertenece. El proceso que realiza la SVM consiste en el mapeo de los datos originales a un espacio de dimensionalidad más alta, a través de una función kernel (función matemática que transforma los datos a un espacio dimensional mayor al original, ϕ) y allí se realiza la clasificación construyendo un hiperplano de separación óptimo Cortes and Vapnik (1995).

Inicialmente fueron diseñados para clasificación pero con la introducción de

la función de pérdida, éstas pueden ser usadas para tareas de regresión Smola and Scholkopf (2004) (las SVM para regresión son llamada Regresión de Vectores de Soporte o SVR por sus siglas en inglés). Así como las SVMs, las SVRs requieren el ajuste de los parámetros libres de la función kernel, C (parámetro positivo que especifica el usuario y que controla el equilibrio entre la complejidad de la máquina y el número de puntos no separables por un hiperplano), γ (radio de la función kernel) y ε (penalización de la función de pérdida of loss function).

1.4 Medidas de rendimiento

Para verificar el rendimiento, existen varias medidas como el Coeficiente de correlación de Pearson R, Coeficiente de correlación de múltiple determination R^2, Raíz del Error Cuadrático Medio $RMSE$, Coeficiente de correlación de múltiple determination R^2_{pred} predictivo, entre otras. Si estas medidas superan un umbral (según lo establecido en modelos QSAR) se acepta el modelo de regresión, por ejemplo, para R, R^2 y R^2_{pred} los valores deben ser mayores a 0.6, 0.5 y 0.5, respectivamente Kiralj and Ferreira (2009), Pratim Roy et al. (2009). A continuación se muestra cómo se calculan dichas métricas:

- Coeficiente de correlación de Pearson R (conjunto de prueba)

$$R_{ext} = \frac{\sum_i (Y_{exp} - \overline{Y_{exp}}) * (Y_{pred} - \overline{Y_{pred}})}{\sqrt{\sum_i (Y_{exp} - \overline{Y_{exp}})^2} * \sqrt{\sum_i (Y_{pred} - \overline{Y_{pred}})^2}} \tag{1}$$

- Coeficiente de correlación de múltiple determination R^2 (conjunto de prueba)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_{exp} - Y_{pred})^2}{\sum_{i=1}^{n} (Y_{exp} - Y_{ave})^2} \tag{2}$$

- Raíz del Error Cuadrático Medio $RMSE$

$$RMSE = \sqrt{\sum_i \frac{(Y_{exp} - Y_{pred})^2}{M}} \tag{3}$$

- Coeficiente de correlación de múltiple determination R^2_{pred} predictivo (conjunto de prueba)

$$R^2_{pred} = \frac{\sum_i (Y_{exp} - Y_{pred})^2}{\sum_i (Y_{exp} - \overline{Y_{cal}})^2} \tag{4}$$

donde M es el número de muestras (en el conjunto de entrenamiento o validación externa); i es el índice de suma y también del i-avo individuo de la muestra; Y_{exp} es el valor experimental de **y** o salida real; Y_{cal} es el dato predicho de **y** en el conjunto de entrenamiento o calibración; Y_{pred} es el valor predicho de **y**, en el conjunto de validación externo.

Así mismo, el conjunto de entrenamiento corresponde a aquellos datos que se emplearán para crear el predictor, el grupo de prueba son los elementos que se usan para la validación cruzada/bootstrapping y/o selección del modelo y que no están incluidos en el grupo anteriormente indicado. El conjunto de validación son los elementos usados para medir el rendimiento final.

1.5. Trabajos previos

Para la predicción de la actividad antimicrobiana de péptidos (con las secuencias CAMELs), se han empleado diferentes algoritmos y descriptores en la literatura. Por ejemplo, en el estudio llevado a cabo por Zhou et al. Zhou et al. (2010), se usaron cerca de 1500 descriptores (que luego de un preprocesamiento se redujeron a 711), en conjunto con Algoritmos Genéticos (AG), Optimización por enjambres de partículas (PSO, por su siglas en inglés) y Máquinas de Soporte Vectorial para regresión (MSV). Sin embargo, en la fase de evaluación del modelo sólo se usaron dos métricas (R y $RMSE$) y de acuerdo a la literatura, es conveniente usar otras como R^2 y R^2_{pred} Kiralj and Ferreira (2009).

En otro estudio reportado por Borkar et al Borkar et al. (2013), se usó Función Genética de Aproximación y Mínimos Cuadrados Parciales (PLS, por sus siglas en inglés) junto con 43 descriptores, sin embargo, el desempeño real del modelo no es claro debido a que los resultados de las métricas reportadas, están basadas en la predicción de la actividad en escala logarítmica. Por otro lado, en el trabajo de Wang et al Wang et al. (2012), aunque los resultados con el entrenamiento son buenos, el desempeño con el grupo de validación es muy bajo. En el trabajo de Torrent et al Torrent et al. (2011) se crea un modelo usando 7 descriptores, pero la evaluación del modelo sólo se limita a dos métricas que que no garantizan el buen funcionamiento del modelo Tropsha (2010).

Un aspecto importante en algunos de estos trabajos es que el conjunto de entrenamiento y validación son explícitos (sólo se realiza una división y en cada caso se indica qué datos se utilizan en cada conjunto), lo que demuestra que los resultados están sesgados. En nuestro trabajo se realizó remuestreo, donde se divide el conjunto de datos varias veces para asegurar la robustez estadística de los resultados.

2. Predicción de la actividad antimicrobiana de péptidos

En este capítulo se propone una metodología que integra aprendizaje y selección de características junto con distintas métricas y estrategias de evaluación de los modelos predictivos, para proporcionar bases sólidas para apoyar la toma de decisiones e inversión, con los recursos de disponibles en laboratorio. Ésto incluyó stacked autoencoders, algoritmos genéticos, curvas de aprendizaje, pruebas de permutación, etc. La finalidad con esta metodología es obtener modelos predictivos con buenos rendimientos y que sean a su vez estadísticamente estables, cuando se utilizan métodos de aprendizaje de máquina para predecir la actividad de péptidos antimicrobianos.

2.1. Materiales y métodos

2.1.1 El conjunto de datos y descriptores

El conjunto de datos empleado en este primer capítulo, corresponde a 101 secuencias de péptidos antibacterianos con tamaño de 15 aminoácidos. Estas secuencias se derivaron de la fusión del C y N terminal de las secuencias de péptidos naturales Cecropin y Melittin. Además éstos péptidos han sido probados contra diversas cepas de microorganismos (gram positivos y gram negativos) y su actividad fue reportada como la potencia media antibiótica contra éstos Cherkasov and Jankovic (2004).

A partir de la secuencia de aminoácidos, es posible codificar información numérica que representa propiedades fisico-químicas (descriptores) del péptido. En este caso, se usaron las propiedades descritas por Zhou et al. Zhou et al. (2010), donde diferentes grupos de descriptores fueron extraídos de la estructura primaria de los péptidos, usando la herramienta web denominada PROFEAT

Rao et al. (2011). Esta herramienta calcula diez grupos de de descriptores mostradas en la Tabla 1, donde AllDesc es el conjunto total de propiedades disponibles. Debido a problemas técnicos con el sitio web de PROFEAT, en su lugar se empleó *propy* (disponible como una librería de Python Cao et al. (2013)) para calcular los diez descriptores mencionados anteriormente.

Tabla 1: Diez grupos de descriptores calculados a partir del conjunto de datos. La segunda y tercera columna representa el número de descriptores antes y después del preprocesamiento, respectivamente

Descriptores	Inicial	Final
Composición dipéptido (Ddcd)	400	106
Autocorrelación normalizada MoreauBroto (Dnmba)	240	112
Autocorrelación Moran (Dmad)	240	112
Autocorrelación Geary (Dgad)	240	112
Composición, transición y distribución (Dctd)	147	147
Sequence order coupling number(Dsoc)	20	20
Quasi sequence order (Dqso)	50	46
Composición Pseudoaminoácidos tipo I (Dpaac)	30	23
Composición Pseudoaminoácidos tipo II (Dapaac)	30	23
Todos los descriptores (AllDesc)	1517	730

Adicionalmente, teniendo en cuenta el orden de los aminoácidos juega un papel importante para su función Wang et al. (2011), se decidió incluir un grupo llamado Vector Composición de Momento (VCM) Ruan et al. (2005) que mide el orden y la frecuencia de aminoácidos en la secuencia. En total se consideraron 11 grupos de descriptores.

2.1.2 Sparse autoencoders (AE)

Un sparse autoencoder Shin et al. (2013) es una clase especial de red neuronal, usada de forma no supervisada. Típicamente, los métodos de aprendizaje supervisado (tales como redes neuronales) parten de un conjunto de datos de entrada junto con sus respectivas etiquetas o predicciones esperadas, para generar un modelo predictivo (por ejemplo, para predecir la actividad antimicrobiana de los péptidos a partir de los descriptores). A diferencia de éstos, los métodos de aprendizaje no supervisados, como AE, utilizan únicamente los datos de entrada para aprender una nueva representación (sin utilizar las etiquetas).

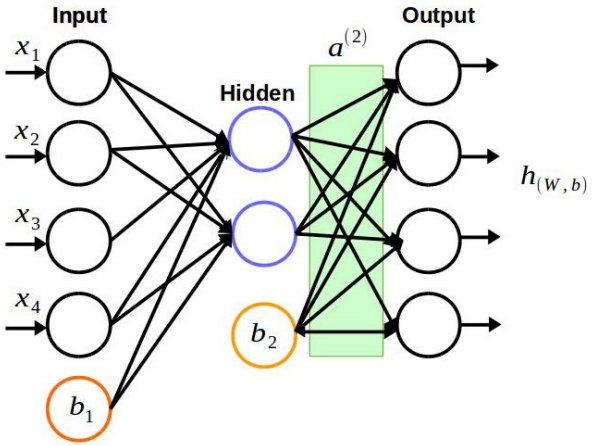

Figura 1: Ejemplo de un sparse autoencoder con 4 neuronas de entrada y dos en la capa oculta. Cabe recalcar que éste es un autoencoder comprimido. Éste contiene (4+1)*2+(2+1)*4 = 22 conexiones

Un AE es una red neuronal simétrica con una capa oculta (Figura 1), es decir, el número de neuronas en la capa de entrada y salida son el mismo. Para cada vector de entrada, la salida de la red se aproxima a los datos de entrada, minimizando el error entre éstos, y el entrenamiento se lleva a cabo de manera similar a una red neuronal. De esta manera, si el entrenamiento tiene éxito en la reconstrucción de los datos de entrada en la capa de salida, la capa oculta contendrá una nueva representación de los datos de entrada, que será más compacto o más sparse si la capa oculta tiene menos o más neuronas que la capa de entrada, respectivamente.

La activación de la red neuronal en la capa oculta, se da producto de la combinación lineal del vector de entrada x:

$$a^{(2)} = f\left(W^{(1)} * x + b^{(1)}\right) \tag{5}$$

donde $W^{(1)}$ es el vector de pesos, $b_{(1)}$ es el bias o término intercepto y f es la función sigmoide donde $f = \frac{1}{1+e^a}$. En la capa de salida, la activación se da como:

$$h_{W,b}(x) = f\left(W^{(2)} * a^{(2)} + b^{(2)}\right). \tag{6}$$

$W^{(2)}$ es el vector de pesos en la capa de salida, $b_{(2)}$ es el bias o término intercepto y f es la función sigmoide. El error de la red al reconstruir la entrada en la salida está dada por la función de costo $J(W,b)$, que es típicamente

minimizada a través del método del gradiente descendente:

$$J(W,b) = \frac{1}{2}\|h_{W,b}(x) - y\|^2 \tag{7}$$

Una característica interesante de los AE es que el número de neuronas en la capa oculta puede ser más pequeño o grande que la capa de entrada. Si es menor, la red comprimirá la información de forma similar que el método de Análisis de Componentes Principales (PCA, por sus siglas en inglés). Si es mayor, la red aprende una representación más distribuida o sparse, en el sentido de que más neuronas se usan para representar la misma información en la capa de entrada. En este caso, el interés se centra en forzar a la red a que active un pequeño número de neuronas de la capa oculta en cada entrada, que producen una representación sparse de los datos y, por lo tanto, obliga a cada neurona a especializarse para detectar un patrón de entrada diferente.

Con el fin de lograr ésto, una restricción de esparcidad se incluye en la función de costo $J(W,b)$ que controla cuántas neuronas son activadas:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{i=1}^{c2} KL(\rho \parallel \hat{\rho_j}) \tag{8}$$

donde β es el peso que penaliza la esparcidad, $c2$ el número de neuronas en la capa oculta, ρ se denomina parámetro de esparcidad (que para este trabajo se empleó un valor igual a 0.05), $\hat{\rho_j} = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right]$ es el promedio de la activación de las neuronas en la capa oculta. KL es la divergencia de Kullback-Leibler:

$$KL(\rho \parallel \hat{\rho_j}) = \rho\, log\, \frac{\rho}{\hat{\rho_j}} + (1-\rho) log\, \frac{1-\rho}{1-\hat{\rho}_j} \tag{9}$$

Los parámetros $W^{(1)}$,$b^{(1)}$,$W^{(2)}$,$b^{(2)}$ se optimizan con el algoritmo de back-propagation y L-BFGS Liu and Nocedal (1989).

2.1.3 Stacked Autoencoders (SAE)

Un stacked autoencoder es una red neuronal con dos o más capas ocultas de autoencoders, cuyo aprendizaje se realiza de forma no supervisada. La idea principal con SAE es capturar características de orden superior a partir de los

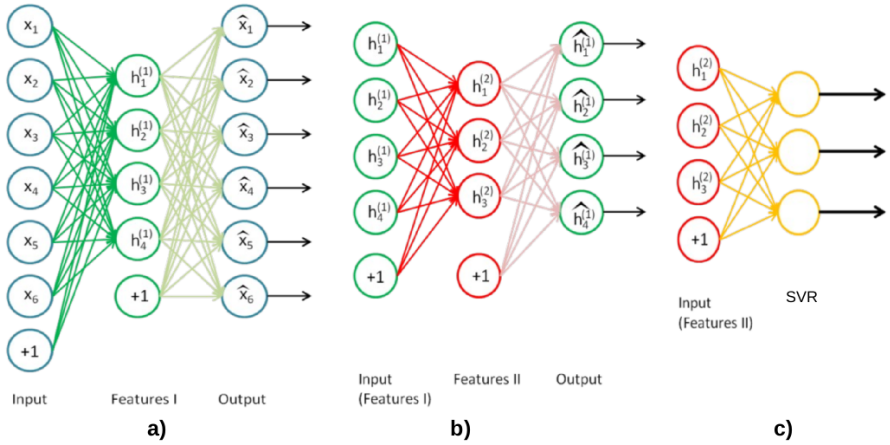

Figura 2: Stacked autoencoder con 2 capas ocultas. a) Entrenamiento en la primera capa oculta. b) Entrenamiento en la segunda capa. c) Representación final que se usa en la SVR. Figura tomada de Ng (2009)

datos. El entrenamiento se lleva a cabo utilizando el enfoque llamado *greedy-wise*, es decir, cada capa oculta es entrenada por separado y la salida de cada uno se utiliza como entrada para la siguiente capa Ng (2009). Por ejemplo, para entrenar un stacked autoencoder con dos capas ocultas, primero se crea y entrena un autoencoder y sólo se almacena la función de activación principal $h^{(1)}$ (ver Figura 2, a) luego del entrenamiento. A continuación, se toma la función de activación del primer autoencoder, para usarse como una nueva representación de los datos de entrada en el segundo autoencoder, donde luego se realiza el proceso de entrenamiento nuevamente (ver Figura 2, b). Al final, el resultado de esta segunda capa $h^{(2)}$ es la representación final de los datos.

2.1.4 Curvas de aprendizaje (LC)

Las curvas de aprendizaje son una representación gráfica del desempeño de un método de aprendizaje de máquina, que se entrena incrementando el número de datos. Para una métrica dada, el rendimiento se mide en el entrenamiento y prueba del conjunto de datos, aumentando el tamaño de la división entre éstos subconjuntos, típicamente, del 10 % al 90 % en un paso de 10 %. Este proceso se lleva a cabo n veces muestreando con reemplazo y luego, se calcula el promedio y desviación estándar para cada porcentaje particular de entrenamiento/prueba. Las curvas de aprendizaje permiten evaluar la capacidad de generalización del método, identificando escenarios de alta varianza (sobreajuste) o alto sesgo (underfitting) Hastie et al. (2009) y soportan la toma de decisiones para mejorar

el rendimiento de los experimentos (adquirir más datos, reducir o aumentar la complejidad del algoritmo, etc.)

2.1.5 Prueba de permutación (PT)

La prueba de permutación es un procedimiento para evaluar la confiabilidad del error del modelo usando una noción de significancia estadística, es decir, es una medida de probabilidad de que el error observado fue producto de la casualidad. Un predictor significativo debería rechazar la hipótesis nula de que las características y las etiquetas son independientes con un p-value pequeño Golland et al. (2000).

Este procedimiento toma el conjunto original de datos, se permuta aleatoriamente las etiquetas, luego se entrena el predictor y se calculan las métricas. Este proceso se repite k veces y se obtiene una distribución de los valores de las métricas. El p-value se calcula de acuerdo a la definición 1 en Ojala and Garriga (2010):

$$p = \frac{|\,\{D' \in \hat{D} : e(f,D') \leq e(f,D)\} + 1\,|}{k+1} \tag{10}$$

donde $D = (X_i, y_i)_{i=1}^{n}$ son los datos con sus respectivas etiquetas, f es la función aprendida por el algoritmo de predicción, $\hat{D}$ es el conjunto de los k aleatorizados del conjunto original, $e(f,D^{'})$ es el error del predictor con los etiquetas aleatorias, $e(f,D)$ es el error con los datos originales.

2.1.4 Flujo de trabajo

El flujo de trabajo llevado a cabo en esta primera parte, está compuesto de cinco etapas:

1. **Preprocesamiento**: todos los descriptores son preprocesados (1) estandarizándolos con media cero y desviación estándar igual a uno; y (2) removiendo aquellos que tenían el mismo valor en todos los péptidos (su desviación estándar era de cero). Esta fase es representada en la Figura 3,3.

2. **Aprendizaje no supervisado de características**: Diferentes configuraciones de AE y SAE se entrenan y se ejecutan sobre los datos preprocesados,

produciendo una nueva representación (Ver Figura 3,11-13 demarcado con verde).

3. **Selección de características**: Se utilizó algoritmos genéticos para seleccionar los descriptores en cada grupo (Ver Figura 3,10 demarcado con magenta).

4. **Predicción supervisado**: Diferentes configuraciones de Regresión de Vectores de Soporte se ejecutan sobre los descriptores, para predecir la actividad antimicrobiana de los péptidos (Ver Figura 3,5-6 señalado con azul).

5. **Evaluación del rendimiento**: se verifica el rendimiento de los modelos usando diferentes métricas y estrategias (Ver Figura 3,7-9 señalado con naranja).

2.2. Montaje experimental

Partiendo de los 11 grupos de descriptores de cada péptido obtenidos con *propy* y luego de ser preprocesados, se realizaron cinco montajes experimentales generales (Figura 3), como se describió anteriormente. Las cinco configuraciones fueron las siguientes:

Original: se ejecutó la SVR directamente sobre cada grupo de descriptores sin aplicar aprendizaje de características. El propósito de éste es obtener una línea de base y un punto de comparación con otras configuraciones.

GA: se optimizó cada conjunto de descriptores usando Algoritmos Genéticos (GA, por sus siglas en inglés), como se muestra en la figura 3. Cada cromosoma está representado por un vector binario, donde 1 es la característica seleccionada y 0, lo contrario. Luego, la función objetivo es usada para evaluar cada cromosoma y determinar el mejor. Para este trabajo, la función objetivo fue tomada de Zhou et. al Zhou et al. (2010):

$$fitness = p * RMSE - \frac{(1-p) * n}{N} \tag{11}$$

donde p es el coeficiente de ponderación que controla el equilibrio entre la precisión del modelo de regresión y el número de descriptores seleccionados, $RMSE$ es la raíz del error cuadrático medio, n es el número de descriptores seleccionados y N el total de descriptores después del pre-procesamiento. Durante

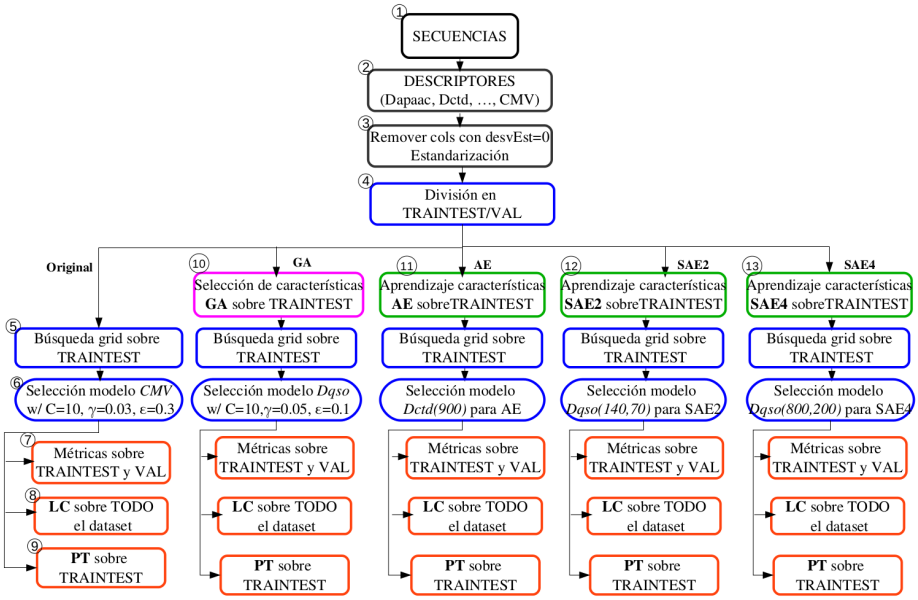

Figura 3: Representación gráfica de la metodología empleada. Los colores representan cada una de las etapas del flujo de trabajo.

el curso de la evolución, los cromosomas están sujetos a la selección, cruce y mutación. Para cada cromosoma, se entrena y prueba una SVR con validación cruzada de 5 grupos, luego de optimizar los parámetros libres, y se calcula el promedio del *RMSE* en esta validación. Un total de 200K SVRs fueron entrenadas y probadas, sólo para este experimento. Finalmente, los parámetros del algoritmo fueron tomados de Shu et al Shu et al. (n.d.), donde la población fue 200, el número de generaciones fue 200, la frecuencia de cruce de 0.5 y la mutación de 0.005.

AE: se entrenaron diferentes configuraciones de autoencoders para aprender un nuevo conjunto de características, que luego fueron usadas con la SVR. En cada configuración de AE, se varió el número de neuronas en la capa oculta entre 20 y 1000 neuronas. Esto permite configuraciones que producen representaciones compactas y sparse, con respecto al número de descriptores en cada grupo. Cuando el número de neuronas en la capa oculta era de entre 20 y 500, éstas fueron variados con un paso de 20, y cuando estaban entre 500 y 1000, se tomó un paso de 50. Ésto dio como resultado 35 configuraciones de AE que se utilizaron para cada grupo de descriptores.

Cada una de estas configuraciones contienen varios miles de conexiones que

necesitan ser entrenadas. Por ejemplo, un AE con 500 neuronas (en la capa oculta) para el grupo descriptor *Ddcd* (106 descriptores) contiene alrededor de 106K conexiones (ver Figura 1). De esta manera el tamaño de los AE empleados osciló entre 800 conexiones (para el AE con los 20 descriptores del grupo Dsoc y 20 neuronas en la capa oculta) y 1,46 millones de conexiones (para All-Desc y 1000 neuronas en la capa oculta).

SAE2: Para cada configuración AE se crearon dos capas de stacked auto-encoder, adicionando una capa oculta con la mitad de neuronas, produciendo otras 35 configuraciones. Como se ha explicado, cada configuración fue entrenada con layer-wise. Los tamaños de las configuraciones SAE2 oscilaron entre 800 conexiones y 1,6 millones.

SAE4: este montaje es similar a SAE2 pero el número de neuronas en la segunda capa oculta se obtuvo dividiendo la cantidad de neuronas de la primera por cuatro, produciendo otras 35 configuraciones. Los tamaños de las configuraciones SAE4 oscilaron entre 600 conexiones y 1.1 millones.

Como el número de descriptores empleados en esta sección fueron 11, en total se realizaron 1177 configuraciones experimentales (385 para **AE**, **SAE2**, **SAE4**, 11 para **Original** y 11 para **GA**). Cabe aclarar que con el nuevo grupo, VCM, se realizaron los mismos montajes (Original, GA, AE, SAE2, SAE4).

2.3. Validación y aprendizaje supervisado

Para validar los resultados de los modelos desarrollados se empleó dos etapas, la primera (denominada Mets) que emplea el mismo esquema y métricas reportadas en la literatura y en la segunda fase (llamada PT-LC), se aplican estrategias como curvas de aprendizaje, prueba de permutación y bootstrapping para determinar la estabilidad estadística de los modelos. En este caso el aprendizaje es supervisado, con lo cual los datos se dividieron en un subconjunto para entrenamiento/prueba y otro para validación de acuerdo a lo indicado en Zhou et al. Zhou et al. (2010) y se optimizó los parámetros de la SVR.

Para ésto, se usó la metodología planteada por Chang et al Chang and Lin (2011), a partir de la cual se construye una malla variando (C, γ, ε) y para cada combinación de parámetros, se toma el conjunto de entrenamiento/prueba, se

entrena y se valida con validación cruzada (5 folds) y con el promedio obtenido con cada combinación (C, γ,ε), se escoge el máximo R^2 (Coeficiente de Correlación Cuadrático). La malla construida fue producto de variar los parámetros libres C (10 a 72.5 con un paso de 2.5), γ ($10^{-1,5}$ a $10^{0,5}$ variando la potencia en un paso de 0.25), ε (0.1 a 0.9 con un paso de 0.1), con lo cual se obtuvo 1872 combinaciones de parámetros que se realizaron con cada configuración descrita en la Sección 2.2. Por tanto, se entrenaron 4'382.752 SVRs con validación cruzada y se seleccionó una configuración, para cada una de las 1177 configuraciones de **Original**, **GA**, **AE**, **SAE2** y **SAE4**.

Acto seguido, con la mejor combinación de parámetros (C, γ, ε) para cada configuración experimental, se entrenó una SVR con el subconjunto de entrenamiento completo (sin validación cruzada) y se probó con el de validación para la obtener el desempeño final. Las métricas de desempeño usadas para el conjunto de validación fueron Raíz del Error Cuadrático Medio ($RMSE_{ext}$), el Coeficiente de Correlación de Pearson (R_{ext}), el Coeficiente de Correlación Cuadrático (R^2_{ext}) y R^2_{pred} (R^2 predictivo, Pratim Roy et al. (2009)) (Ver figura 3,7).

Luego, en una segunda etapa, se tomaron los grupos de descriptores con el mejor rendimiento en cada configuración y se aplicaron las siguientes estrategias:

LC: la curva de aprendizaje (LC) es calculada para todo el conjunto de datos usando la SVR con la mejor combinación de parámetros obtenida en las configuraciones experimentales descritas anteriormente (Ver figura 3,8). Este proceso se lleva a cabo 30 veces muestreando con reemplazo.

PT: la prueba de permutación (PT) se aplica sobre diferentes conjuntos de entrenamiento/prueba (éstos se toman cuando se aplica bootstrapping sobre el conjunto total de datos y se dividen a su vez en entrenamiento/prueba, r veces). Este proceso se lleva a cabo 10 veces (r=10) y se permutan aleatoriamente las salidas 100 veces (k=100) para cada división r (Ver figura 3,9). Esta prueba se aplica para la mejor configuración en cada experimento.

2.4. Resultados

2.4.1. Resultados etapa Mets

La metodología mostrado anteriormente se diferencia de la mayoría de los estudios usados en la predicción de péptidos antimicrobianos (Shu et al. (n.d.), Hemmateenejad et al. (2011), Lin et al. (2008), Zhou et al. (2010), Wang et al. (2012), Borkar et al. (2013), Torrent et al. (2011)), en que los descriptores se aprenden de forma automática, en una tarea de aprendizaje no supervisada. La tabla 2 resume los resultados reportados en la literatura y los que se obtuvieron en este trabajo (se muestran en las cinco líneas al final de la tabla, junto con el grupo de descriptor y la configuración AE o SAE con la que se obtuvieron).

Tabla 2: Comparación de resultados para diferentes algoritmos usados para la predicción de la actividad de péptidos antimicrobianos

Método	R_{ext}	$RMSE_{ext}$	R^2_{ext}	R^2_{pred}	**Ref**
GA-SVM	0.78	1.39	-	-	Zhou et al. (2010)
PSO-GA-SVM	0.9	0.96	-	-	Zhou et al. (2010)
STR-MLR	-	-	0.326	-	Wang et al. (2012)
G/PLS	0.8	-	0.67	0.64	Borkar et al. (2013)
ANN	-	-	0.72	-	Torrent et al. (2011)
Original (CMV+SVR)	0.87	1.10	0.74	0.74	Este trabajo
GA (Dqso+GA+SVR)	0.92	0.85	0.84	0.85	Este trabajo
AE (Dctd(900)+SVR)	0.9	1.10	0.739	0.74	Este trabajo
SAE2 (Dqso(140,70)+SVR)	0.96	0.864	0.841	0.842	Este trabajo
SAE4 (Dqso(800,200)+SVR)	0.97	0.845	0.848	0.85	Este trabajo

GA = Algoritmos Genéticos, SVM = Máquinas de Soporte Vectorial, PSO = Optimización de Enjambre de Partículas, G/PLS = Mínimos cuadrados parciales, STR = Regresión paso a paso, MLR = Regresión Múltiple Lineal, ANN = Red Neuronal Artificial

En las figuras 4, 5 y 6 se muestran los desempeños de cada grupo descriptor con R^2_{ext}, $RMSE_{ext}$ y R_{ext} respectivamente, junto con el reportado en la literatura (representado por las líneas de colores, según la referencia de la literatura mostrada en la tabla 2). En la configuración Original, es decir, sin aplicar ninguna estrategia de selección o aprendizaje de características, se evidencia que los desempeños son mucho menores que GA, SAE2 o SAE4. Además, se muestra que los descriptores relacionados con el orden de los aminoácidos (Dapaac, Dqso y VCM), ofrecen buenos resultados en este caso.

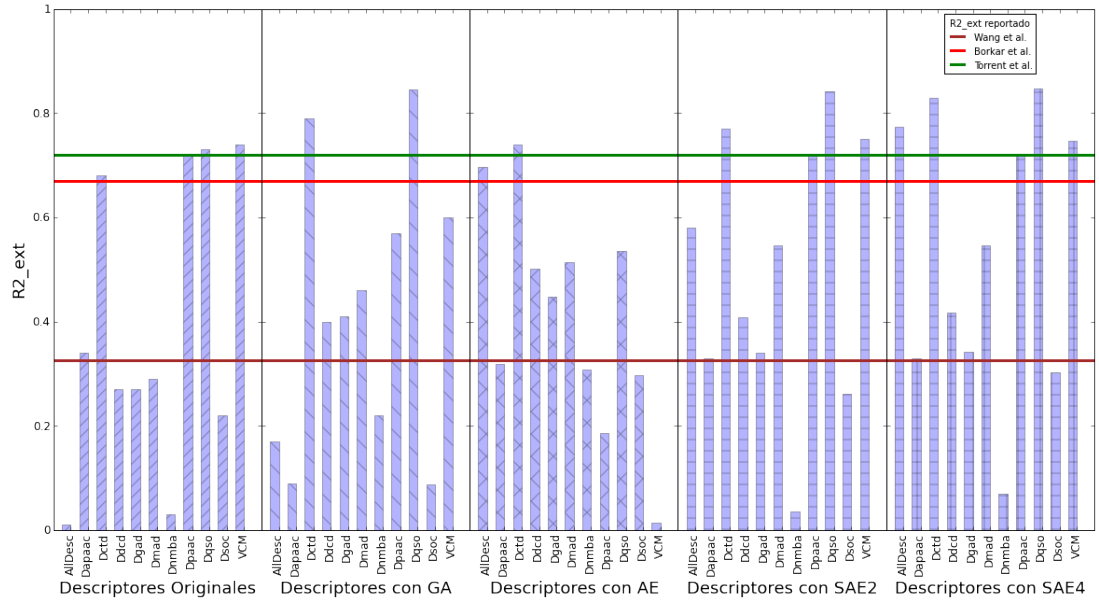

Figura 4: Mejores rendimientos para R^2_{ext} con cada grupo de descriptores, en las cinco configuraciones experimentales. R^2_{ext} mientras más cercano a 1, es mejor el rendimiento

En la configuración GA, la mayoría de los rendimientos aumentaron con respecto a Original, por ejemplo como en el grupo Dqso (que mostró el mejor resultado), aunque en otros casos como Dapaac se redujo (esto puede deberse a que, este conjunto globalmente contiene mayor información que se correlaciona mejor con la actividad). Adicionalmente, este resultado superó al reportado por Zhou et al. Zhou et al. (2010) e incluso el tiempo de cómputo que se requirió (48 horas, sin usar ningún esquema de paralelismo, como si se usó en el trabajo de Zhou) fue mucho menor.

En la configuración SAE2 no muestra mejoras significativas con respecto a **SAE4**. Con las configuraciones AE, SAE2 y SAE4, los experimentos tomaron cerca de 40 horas, aunque el tiempo de cómputo para cada conjunto de descriptores y de **AE** o **SAE** varía significativamente en función del número de conexiones de la configuración especificada.
Para la configuración **AE**, el mejor grupo de descriptores fue Dctd (con 147 descriptores originalmente) con 900 neuronas en la capa oculta (este es un autoencoder con 265K conexiones). Sin embargo, **SAE4** obtuvo un mejor desempeño con los grupos Dctd y Dqso, comparados con los mostrados en la literatura.

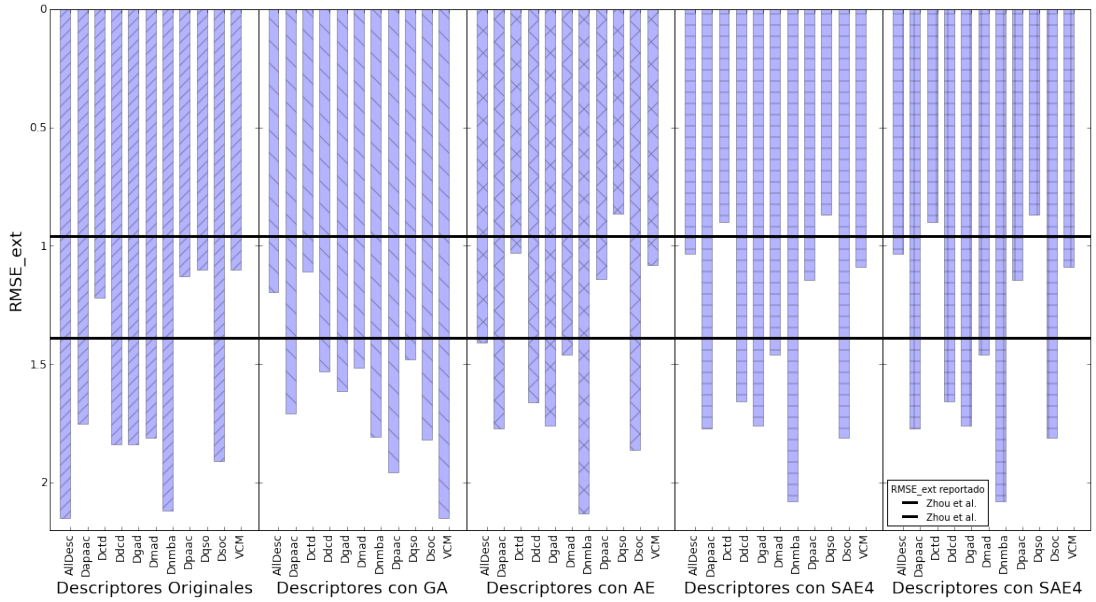

Figura 5: Mejor rendimiento para $RMSE_{ext}$ con cada grupo de descriptores, en las cinco configuraciones experimentales. $RMSE_{ext}$ mientras más cercano a cero, es mejor el rendimiento

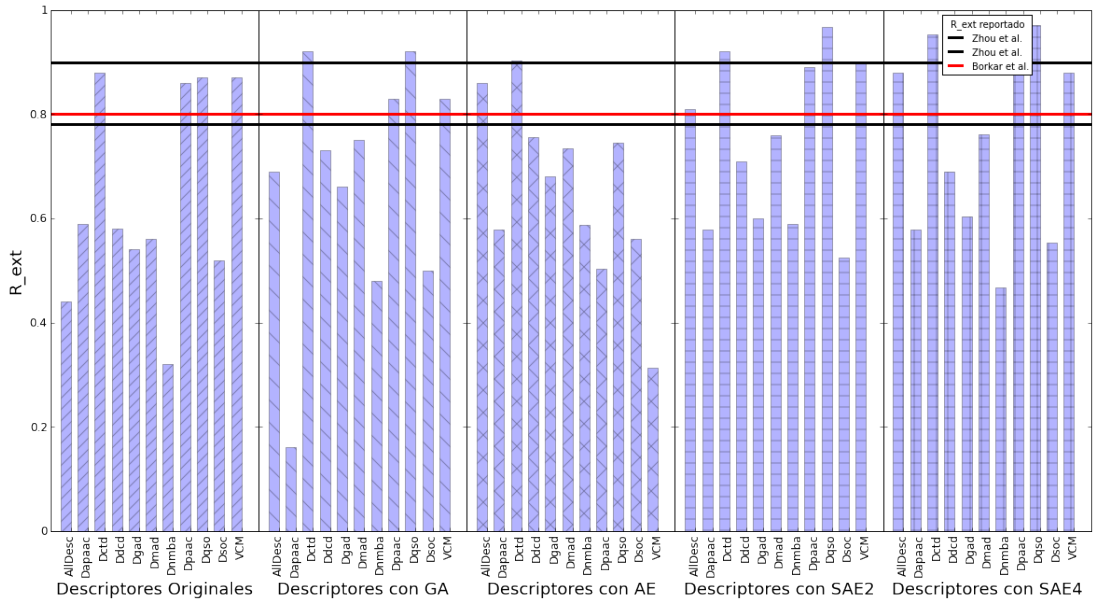

Figura 6: Mejor rendimiento para R_{ext} con cada grupo de descriptores, en las tres configuraciones experimentales. R_{ext} mientras más cercano a 1, es mejor el rendimiento

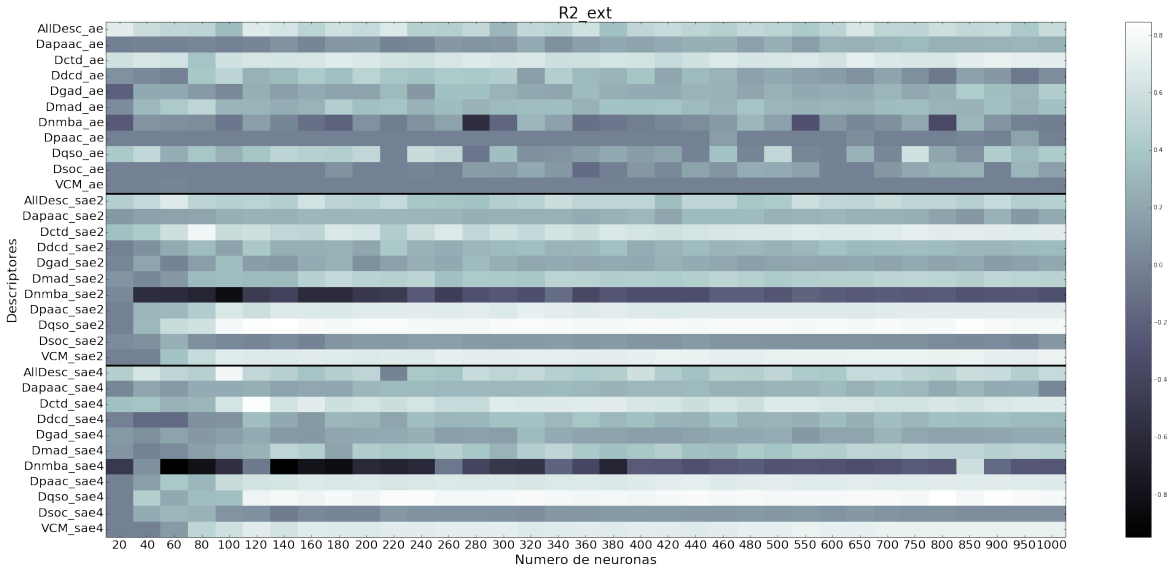

Figura 7: Representación gráfica de los resultados de los autoencoder y stacked autoencoder cuando se varía el número de neuronas en la capa oculta. El mejor resultado para AE fue Dctd con 900 neuronas, SAE2 con (140,70) y SAE4 con Dqso (800,200)

En la figura 7 se muestran los resultados obtenidos para todas las configuraciones de autoencoder y stacked autoencoder, con cada conjunto de descriptores para la métrica R^2_{ext}, variando el número de neuronas en la primera capa oculta. Cabe recordar que para **SAE2**, la segunda capa oculta contiene la mitad de neuronas de la primera y para **SAE4**, la cuarta parte. Los valores altos y bajos de R^2_{ext} están representados con blanco (el mejor puntaje) y azul (peor puntaje), respectivamente, por ejemplo, se puede observar cómo los grupos de descriptores Dqso y Dpaac se comportan consistentemente bien en **SAE2** y **SAE4**, caso contrario al del grupo Dnmba y con **AE** el grupo que parece mejor es Dctd.

Cabe destacar que las configuraciones con más neuronas en las capas ocultas parecen funcionar mejor (para cada fila, la puntuación en el lado derecho tiende a ser más clara), lo que favorecería a los AEs y SAEs que aprenden una representación dispersa en contraposición a los de aprendizaje más compacto (compresión) (Ver Figura 7).

Por último, con el fin de mostrar una interpretación de las características aprendidas, se compararon las correlaciones entre los descriptores originales de cada grupo y las obtenidas con la mejor configuración SAE en cada caso. Los

resultados pueden observarse en la Tabla 3 donde cada imagen es la matriz de correlación entre $n \times n$ descriptores (mostrado en escala de grises) y en la cual, cada pixel representa la correlación entre las correspondientes variables.

Así, la dependencia completa entre variables es representada por una diagonal blanca rodeado por un fondo negro. En este caso, puede observarse que en general, las nuevas características obtenidas a través de SAEs mejoran la independencia de los descriptores originales (al ser más oscuros), como se muestra en la fila 2 de la tabla 3.

Tabla 3: Correlación dentro de cada grupo de descriptores para las configuraciones de SAE, comparadas con la de la representación original. Si el pixel es más oscuro, la correlación es baja, mientras que si es más clara, la correlación es mayor.

-	AllDesc	Dapaac	Dctd	Ddcd	Dgad	Dmad	Dnmba	Dpaac	Dqso	Dsoc
Original										
SAE										
Neuronas	100,25	460,230	120,30	220,110	100,50	180,90	40,10	320,160	800,200	60,15

2.4.2. Resultados etapa PT-LC

El siguiente paso fue evaluar las configuraciones experimentales mostradas en la tabla 2 usando curvas de aprendizaje y la prueba de permutación. Las curvas de aprendizaje se calcularon para la métrica R^2_{ext} en cada configuración experimental; éstas se muestran en la figura 8a, 8b, 8c, 8d y 8e que corresponden a Original, GA, AE, SAE2 y SAE4.

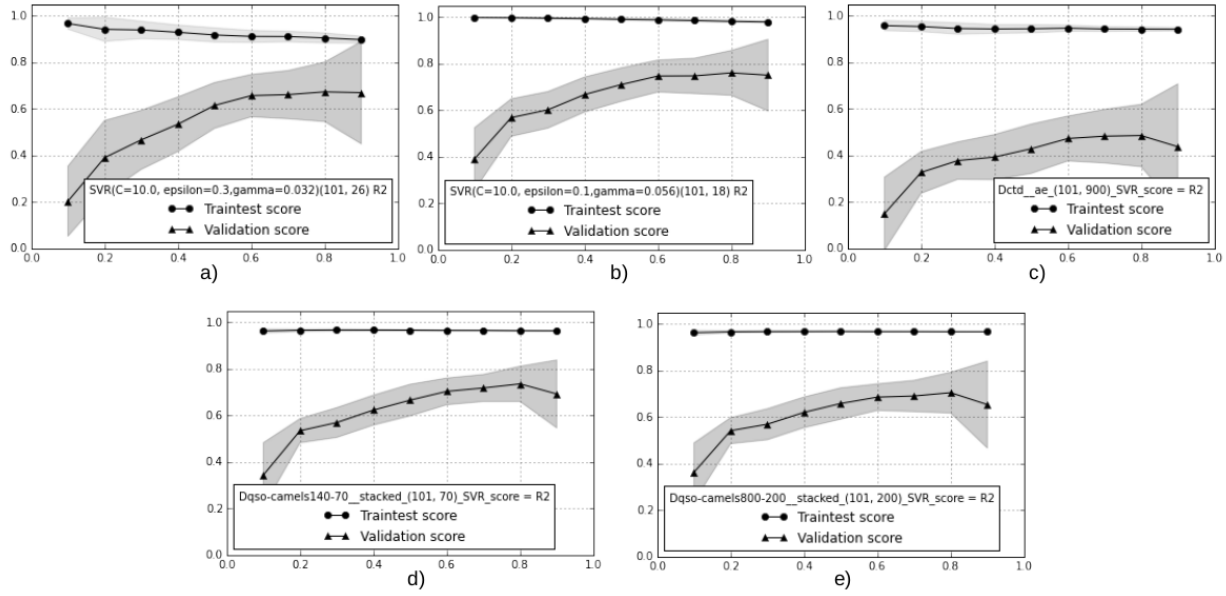

Figura 8: Representación gráfica de las curvas de aprendizaje con los mejores descriptores en cada configuración experimental y el rendimiento corresponde a la métrica R^2_{ext}.

La figura 8b muestra el mejor comportamiento puesto que el rendimiento con el conjunto de validación converge con el de entrenamiento/prueba y con valores altos, mostrando el menor underfitting y sobreajuste (éste se obtuvo con GA). Por lo tanto, se puede afirmar que el rendimiento se estabiliza en 0.84 en la métrica R^2_{ext}. Las figuras 8a, 8d y 8e muestran un desempeño menor puesto que añadir más datos aún no elimina la brecha entre el entrenamiento/prueba y la validación. Esto constituye un claro caso en el que los métodos son incapaces de generalizar los datos no vistos durante el entrenamiento (sobreajuste). Este comportamiento se observa más drásticamente en la figura 8c. Tenga en cuenta que el mejor rendimiento (Figura 4B) se obtiene con GA. Para tratar este problema, probablemente se requiere adicionar más datos. Las figuras 8d y 8e muestran un comportamiento interesante con el conjunto de entrenamiento/prueba, porque el rendimiento es similar cuando se incrementa el conjunto de muestras.

Como paso siguiente, se evaluaron los modelos con la prueba de permutación y los resultados obtenidos se observan en las figuras 9a, 9b, 9c, 9d y 9e para cada configuración experimental para Original, GA, AE, SAE2 y SAE4 respectivamente.

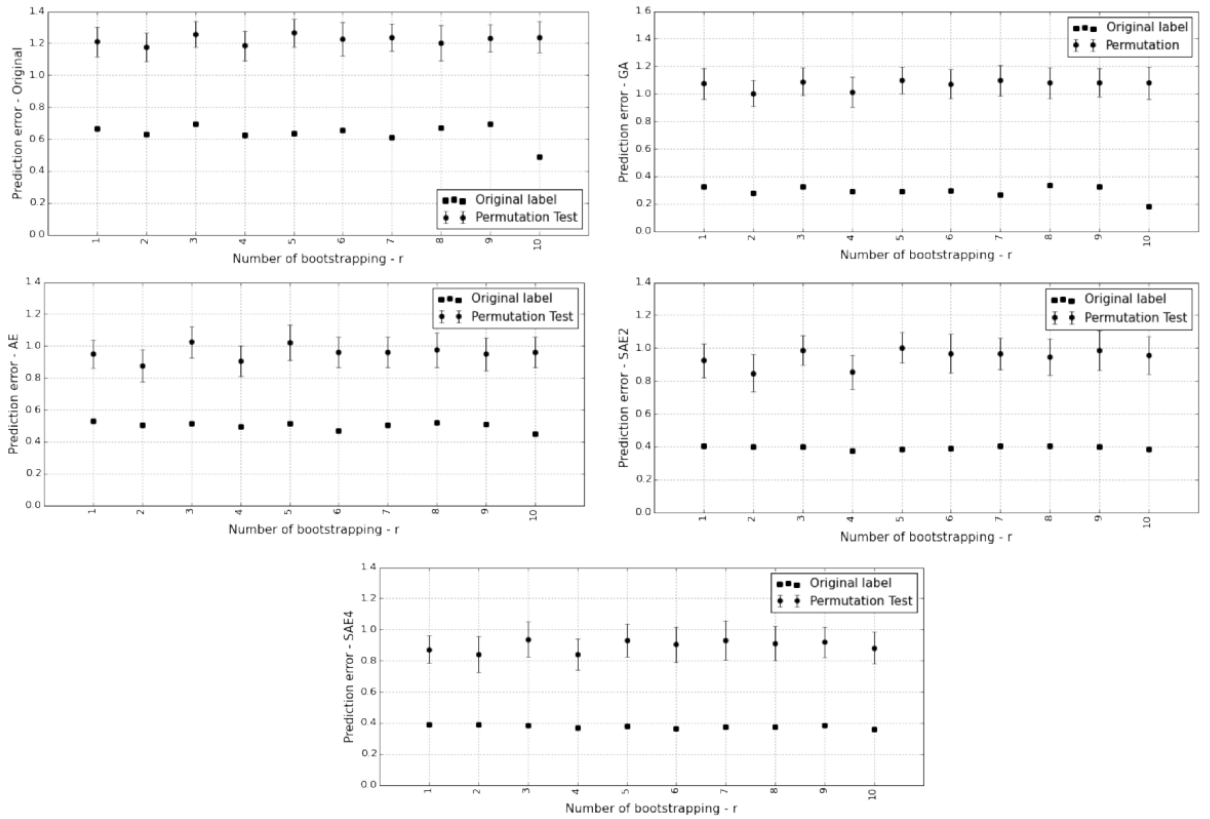

Figura 9: Prueba de permutación para cada iteración del bootstrapping en difererentes configuraciones experimentales.

Esta prueba se realizó para determinar si los resultados mostrados en la primera etapa fueron producto de la casualidad (este problema puede darse cuando hay una alta dimensionalidad, en este caso, un gran número de descriptores y un tamaño de muestra pequeño Golland et al. (2000)). El cuadrado representa el error original para cada bootstrapping y el círculo es el error promedio de la prueba de permutación con las desviación estándar (la línea vertical que cruza el círculo). Si el cuadrado cae debajo del círculo, se rechaza la hipótesis nula, si es el caso contrario (o ambos tienen el mismo error) no se puede rechazar la hipótesis con un nivel de signficancia α=0.01. Estas figuras muestran que la hipótesis nula es rechazada, es decir, el predictor ha encontrado una dependencia real entre los datos y las etiquetas.

Adicionalmente, en la tabla 4 se muestra el *p*-value obtenido para diferentes configuraciones experimentales, donde Err (en las etiquetas originales) representa el error promedio con las etiquetas originales y Desv es la desviación estándar calculada para los 10 bootstrapping. El Err (en la Prueba de permutación) representa el promedio del error con las etiquetas permutadas y Desv es

la desviación estándar calculada para las 1000 permutaciones (r x k, con r = 10, k = 100). Estos resultados indican que los predictores diseñados en cada configuración experimental son significativos bajo la hipótesis nula (con un intervalo de confianza del 95 %).

Tabla 4: Error promedio con las etiquetas originales, prueba de permutación y p-value para cada configuración experimental.

Etiquetas originales		Prueba de permutación	
Configuración	**Err (Desv)**	**Err (Desv)**	**p-val**
Original	0.61 (0.08)	1.15 (0.15)	0.009
GA	0.39 (0.12)	1.20 (0.13)	0.009
AE	0.12 (0.006)	0.57 (0.14)	0.009
SAE2	0.39 (0.09)	0.94 (0.12)	0.009
SAE4	0.19 (0.12)	0.66 (0.15)	0.009

Por otro lado, con el fin de verificar en qué rangos varían los resultados al cambiar aleatoriamente los conjuntos de entrenamiento/prueba y validación, se realizaron 10 bootstrapping (divisiones aleatorias con reemplazo) siguiendo el mismo esquema de la figura 3, cuyos resultados se muestran en la tabla 5. Estos resultados indican que al cambiar el conjunto de entrenamiento, los modelos predictivos como SAE2 y SAE4 van a seguir mostrando buenos desempeños, debido a la poca variabilidad mostrada de éstos. Cabe destacar que el conjunto de validación indicado por Zhou parece sesgar los resultados debido a que éstos son mayores que los rangos obtenidos con el bootstrapping.

Tabla 5: Bootstrapping sobre diferentes configuraciones experimentales usadas en este trabajo. *División Zhou* indica el resultado obtenido por Zhou con el conjunto entrenamiento/prueba y validación especificado. Bootstrapping es el resultado obtenido en este trabajo usando la misma división indicada por Zhou.

Method	R_{ext}		$RMSE_{ext}$	
-	**División Zhou**	**Bootstrapping**	**División Zhou**	**Bootstrapping**
Original	0.87	0.8±0.18	1.10	1.21±0.29
GA	0.92	0.86±0.11	0.85	0.98±0.26
AE	0.9	0.69±0.16	1.10	1.52±0.28
SAE2	0.96	0.86±0.06	0.86	1.07±0.18
SAE4	0.97	0.84±0.09	0.84	1.12±0.25
Zhou et al. (PSO-GA-SVM)	0.9	-	0.96	-

Finalmente, se observó la frecuencia de los parámetros de la SVR para C, γ y ε donde los valores más comunes fueron 10, 0.32 y 0.1 respectivamente, sin embargo, éstos no muestran correlaciones con los resultados de cada métrica.

3. Predicción de la actividad con secuencias GIBec

En esta sección se describe el mismo proceso mostrado en el capítulo anterior, pero usando las secuencias producto de la síntesis realizada en el laboratorio de bioquímica GIBIM. En este caso, se referirá a estas secuencias con el nombre GIBEc.

3.1 Conjunto de datos y descriptores

Las secuencias usadas en esta segunda parte del trabajo corresponde a 41 péptidos sintetizados en el grupo de investigación en bioquímica y microbiología GIBIM. De estos péptidos, un grupo se obtuvo usando un programa realizado dentro del grupo de investigación (una implementación basada en un algoritmo genético) y los restantes, se hicieron a partir de un diseño racional, es decir, se tomó un conjunto de secuencias de péptidos reportados en la literatura (con buena actividad) y se cambiaron algunos aminoácidos, tratando de maximizar ciertas propiedades físicoquímicas. Por otro lado, la actividad de estos péptidos fue medido como la Concentración Mínima Inhibitoria que inhibe el crecimiento de la bacterias en un 50 % (CMI_{50}) y se caracterizan por tener una estructura α hélice, con tamaños entre 17 y 21 aminoácidos. El CMI_{50} mostrado por los pépidos varía entre 0.4 μM y 101 μM.

En el caso de los descriptores, se emplearon los mismos descritos en la sección anterior, pero el número total empleado varió puesto que el grupo CAMELs tiene más secuencias comparado con GIBec. Éstos se muestran en la tabla 6.

Tabla 6: Diez grupos de descriptores calculados a partir del conjunto de datos. La segunda y tercera columna representa el número de descriptores antes y después del preprocesamiento, respectivamente

Descriptores	Inicial	Final
Composición dipéptido (Ddcd)	400	107
Autocorrelación normalizada MoreauBroto (Dnmba)	240	152
Autocorrelación Moran (Dmad)	240	152
Autocorrelación Geary (Dgad)	240	152
Composición, transición y distribución (Dctd)	147	147
Sequence order coupling number(Dsoc)	20	20
Quasi sequence order (Dqso)	60	58
Composición Pseudoaminoácidos tipo I (Dpaac)	30	29
Composición Pseudoaminoácidos tipo II (Dapaac)	30	29
Todos los descriptores (AllDesc)	1407	846

3.2. Resultados

Los resultados de la primera etapa se muestran en la tabla 7, sin embargo, no se reportan otros trabajos en la literatura, puesto que este grupo fue recientemente sintetizado y probado en el grupo de investigación GIBIM.

Tabla 7: Resultados obtenidos para cada configuración experimental

Method	R_{ext}	$RMSE_{ext}$	R^2_{ext}	R^2_{pred}	**Ref**
Original (Dgad+SVR)	0.38	37.14	0.13	0.14	Este trabajo
GA (Dpaac+GA+SVR)	0.58	36.06	0.18	0.19	Este trabajo
AE (Dgad(850)+SVR)	0.55	34.16	0.26	0.27	Este trabajo
SAE2 (Ddcd(340,170)+SVR)	0.6	32.88	0.317	0.326	Este trabajo
SAE4 (Ddcd(480,120)+SVR)	0.59	32.81	0.32	0.329	Este trabajo

En las figuras 10, 11, 12 se muestra el rendimiento de cada descriptor con R^2_{ext}, $RMSE_{ext}$ y R_{ext}, respectivamente. Los resultados mostrados por el experimento Original, no muestran buenos rendimientos en las tres métricas usadas, especialmente en el error que alcanza su máximo cerca al 50. En el caso de los AEs y SAEs, se observa que los resultados mejoraron con respecto al Original, especialmente al usar SAE2, sin embargo, siguen siendo bajos desempeños. Por otro lado, el tiempo empleado para todos los experimentos en esta sección fue de 30 horas, debido a que el número de secuencias del grupo GIBIM es mucho

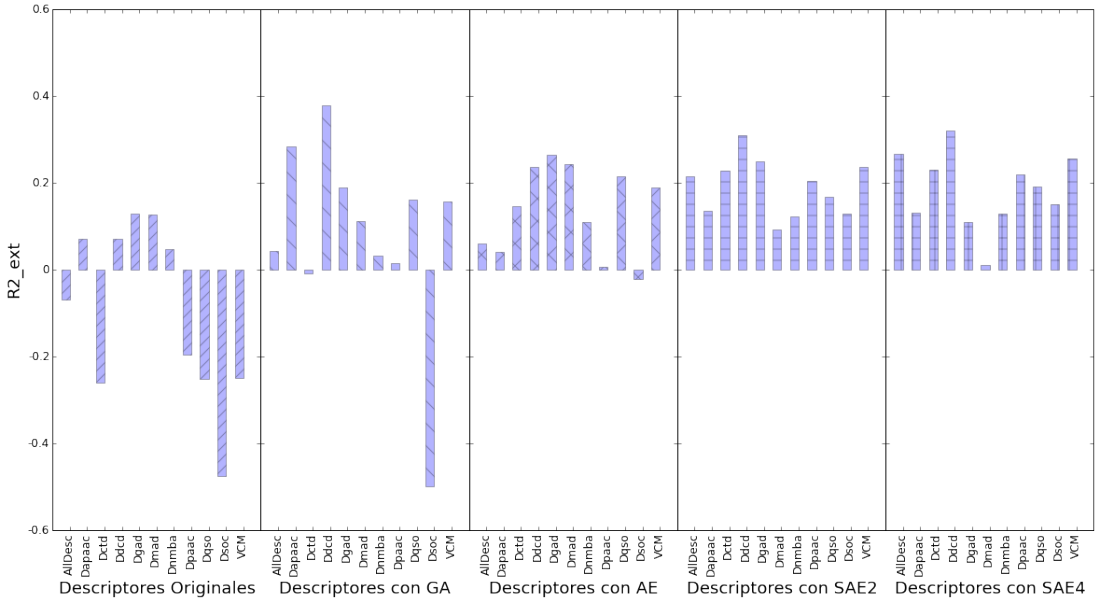

Figura 10: Mejor rendimiento para R^2_{ext} con cada grupo de descriptores en las cinco configuraciones experimentales. R^2_{ext} mientras más alto es mejor

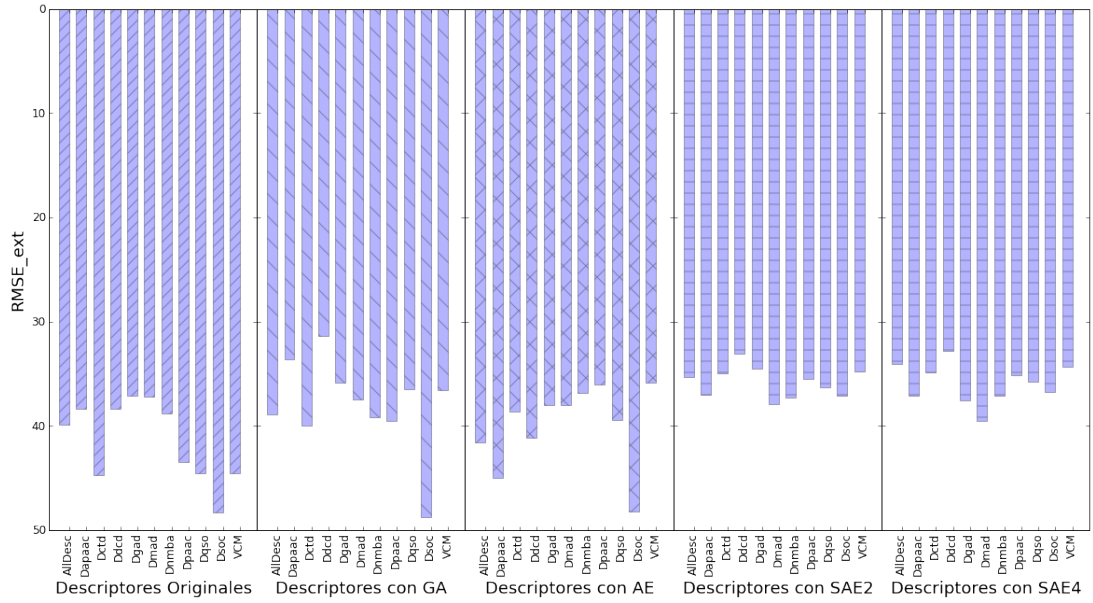

Figura 11: Mejor rendimiento para $RMSE_{ext}$ con cada grupo de descriptores en las cinco configuraciones experimentales. $RMSE_{ext}$ mientras más cercano a cero es mejor

menor que CAMELs.

En la figura 13, se muestran los resultados para todas las configuraciones de algoritmos genéticos, autoencoder y stacked autoencoder para la métrica R^2_{ext} variando el número de neuronas en la primera capa. Un valor alto de R^2_{ext} es representado al acercarse al color blanco mientras que empeora al aproximarse

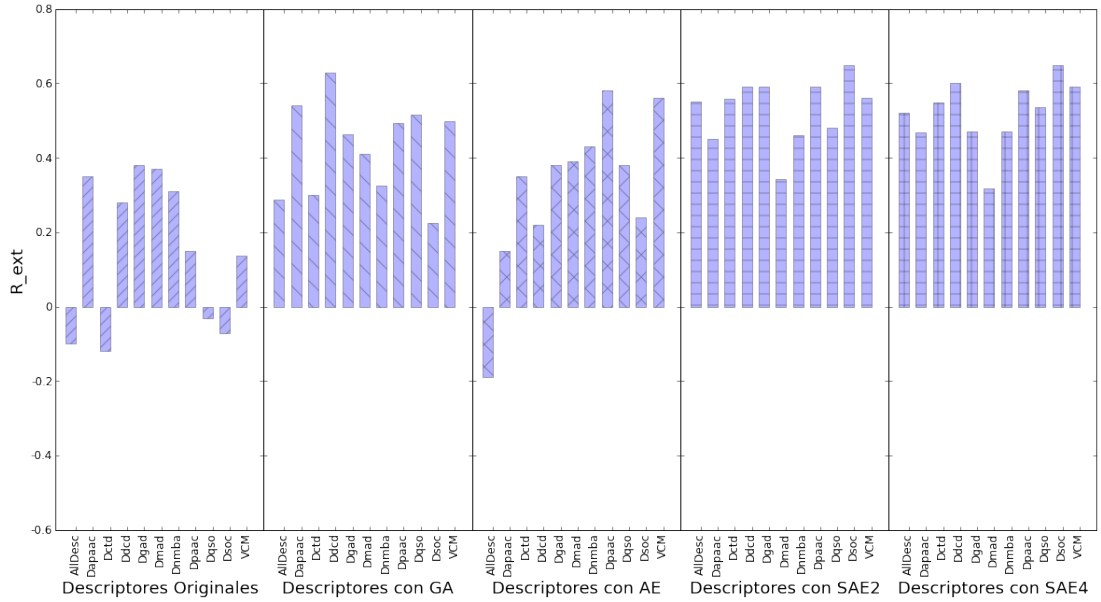

Figura 12: Mejor rendimiento para R_{ext} con cada grupo de descriptores en las cinco configuraciones experimentales. R_{ext} es mejor mientras más cercano a uno

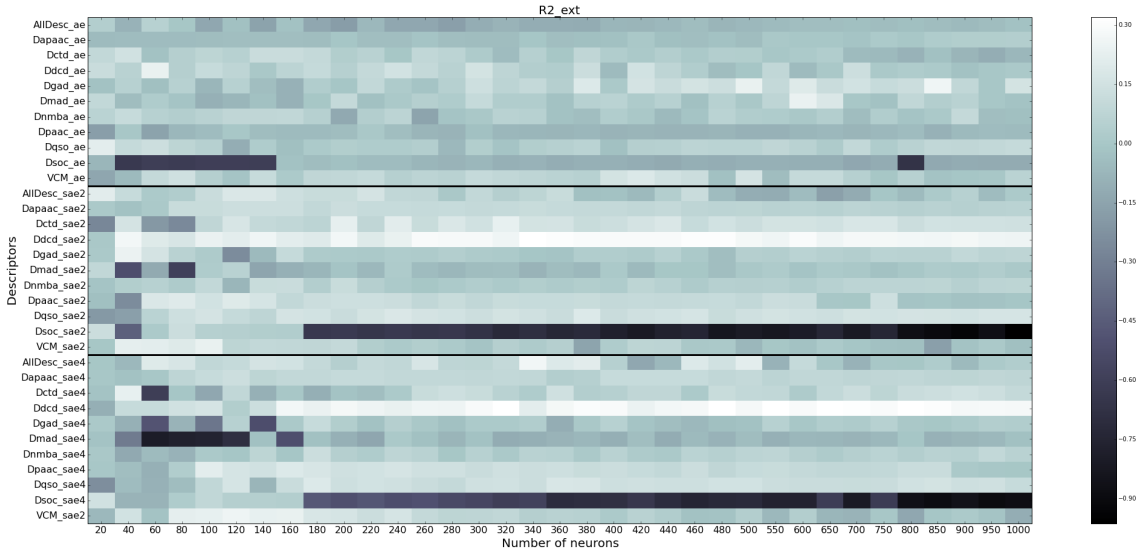

Figura 13: Mejor rendimiento para la métrica $RMSE_{ext}$ con cada grupo de descriptores en las tres configuraciones experimentales. El máximo valor alcanzado fue 0.30 y el mínimo fue -0.9

al azul oscuro.

Se puede observar que el grupo con el mejor rendimiento es Ddcd, cuyos resultados son consistentes en SAE2 y SAE4. En el caso de AE, parece que el grupo Dgad, se desempeña mejor. En este caso, el número de neuronas en la capa oculta no parece afectar los resultados, es decir, que no hay preferencia en la representación sparse ni la compacta (con respecto al número de descriptores).

Adicionalmente, con el fin de dar una interpretación sobre el aprendizaje de características, se compararon las correlaciones entre los descriptores originales y aquellas configuraciones SAE. Estas comparaciones se muestran en la tabla 8, donde la correlación perfecta se representa por la diagonal blanca, rodeada por un fondo negro. Puede observarse que en general, las nuevas características obtenidas a través de SAEs generalmente mejoran la independencia de los descriptores originales al ser más oscuros generalmente, como se muestra en la fila 2 de la tabla. 8.

Tabla 8: Correlación dentro de cada grupo de descriptores para las configuraciones de SAE, comparadas con la de la representación original. Si el pixel es más oscuro, la correlación es menor, mientras que si es más clara, la correlación es mayor.

-	AllDesc	Dapaac	Dctd	Ddcd	Dgad	Dmad	Dnmba	Dpaac	Dqso	Dsoc
Original										
SAE										
Neuronas	340,85	120,60	40,10	650,162	40,20	20,10	180,45	100,25	180,45	20,5

Finalmente, la segunda etapa de la validación no se llevó a cabo porque los resultados en la primera fase para cada una de las métricas R^2_{ext}, $RMSE_{ext}$, R_{ext} y R^2_{pred} no superaron el umbral mínimo para aceptarlo, por ende, se muestra evidencia de que la metodología empleada no funciona con estas secuencias en particular.

4. Conclusiones y recomendaciones generales

En este trabajo se abordó la tarea de predecir la actividad de los péptidos antimicrobianos utilizando una metología que integró selección y aprendizaje de características, curvas de aprendizaje y prueba de permutación. Esta metodología fue aplicada en dos conjuntos de datos: Los CAMELs y GIBec. A partir del proceso realizado para abordar este problema, nuestro aporte se centró en cuatro aspectos:

1. Aprendizaje de características
2. Identificación de problemas como alto sesgo o alta varianza
3. Confiabilidad estadística
4. Requerimientos del problema para aplicar esta metodología

Teniendo los aportes descritos anteriormente, se concluyó que:

- Cuando se usa la nueva representación (aprendida con los autoencoders y stacked autoencoders, de forma no supervisada) como entrada en un método de aprendizaje automático supervisado, se mostró cómo las representaciones aprendidas proporcionan consistentemente resultados satisfactorios en comparación con trabajos recientes.

- Las representaciones sparse parecen ser las preferidas en lugar de las más compactas, ya que probablemente dan una mejor oportunidad de separabilidad de datos en la tarea de predicción supervisada que se realiza después. Asimismo, se muestra cómo las representaciones aprendidas también mejorar la independencia de los descriptores iniciales, reduciendo la correlación entre ellos. Esto sugiere enfoques, probablemente, híbridos donde los expertos elaboran una colección de descriptores base y se complemente

con el proceso de selección y aprendizaje no supervisado de características.

- Por otro lado, al usar las curvas de aprendizaje se evidenció que la configuración AE sufre de overfitting, por lo tanto es necesario incrementar el número de muestras (ésto sujeto a la disponibilidad de recursos en el laboratorio). Esta estrategia se podría implementar con el fin de mejorar los resultados para AE. En el caso de los modelos desarrollados con GA, SAE2 y SAE4 se evidenció que no sufren problemas de alta varianza o alto sesgo, lo que indica que éstos son capaces de generalizar. Esto sumado a la evidencia mostrada por las pruebas de permutación, en la cual los modelos creados encontraron una dependencia real entre descriptores y la actividad, indica que GA, SAE2 y SAE4 (particularmente este último, por los buenos rendimientos mostrados), ofrecen buenos resultados y además son estadísticamente estables.

- También se identificaron grupos de descriptores que se comportan consistentemente mejor, este es el caso del Vector Composición de Momento y Quasi Sequence Order, que indican que el orden, la frecuencia y las propiedades asociadas a éstos en la secuencia, podría ayudar a diseñar mejores péptidos candidatos a sintetizar en el futuro.

- Con el conjunto de secuencias GIBEc, el aprendizaje de características mejoró los resultados con respecto a la configuración experimental Original y GA, sin embargo, los desempeños mostrados no superaron los umbrales establecidos para aceptar los modelos. Esto indica que el esquema metodológico empleado no funciona con el grupo GIBec, debido a: los péptidos empleados tienen una baja identidad entre ellos, por lo cual los descriptores pueden variar considerablemente entre secuencias y dificultar la identificación de patrones. Otro factor que incide es la actividad reportada, puesto que ésta no es reportada como un valor puntual sino un rango de valores posibles donde está el CMI del péptido. Además no se cuenta

con una muestra representativa de secuencias, debido a que el proceso experimental para hacer la síntesis de los péptidos puede tardar semanas y actualmente se está trabajando en el análisis del mecanismo de acción, por ende, no se puede disponer de más datos (por el momento).

Finalmente, esta metodología podría explorarse en otras áreas de predicción de propiedades de proteínas con tareas similares, siempre que el conjunto de datos usado presente alta identidad entre secuencias, la medida a predecir sea un valor puntual y que el tamaño de la muestra sea representativo. Basado en estas condiciones, esta metodología puede ayudar a crear buenos modelos predictivos usando aprendizaje de características para obtener mejores correlaciones entre variables dependientes e independientes, además permite identificar diversos problemas en los predictores y asegura que éstos sean estadísticamente estables.

Bibliografía

Amábile-Cuevas, C. F. (2010). Global Perspectives of Antibiotic Resistance, *in* A. d. J. Sosa, D. K. Byarugaba, C. F. Amábile-Cuevas, P.-R. Hsueh, S. Kariuki and I. N. Okeke (eds), *Antimicrobial Resistance in Developing Countries*, Springer New York, New York, NY, chapter 1, pp. 3–13.

Borkar, M. R., Pissurlenkar, R. R. S. and Coutinho, E. C. (2013). HomoSAR: Bridging comparative protein modeling with quantitative structural activity relationship to design new peptides, *Journal of Computational Chemistry* **34**(30): 2635–2646.

Cao, D. S., Xu, Q. S. and Liang, Y. Z. (2013). Propy: A tool to generate various modes of Chou's PseAAC, *Bioinformatics* **29**(7): 960–962.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology* **2**(3): 1–27.

Cherkasov, A. (2005). 'Inductive' Descriptors: 10 Successful Years in QSAR, *Current Computer - Aided Drug Design* **1**(1): 21–42.

Cherkasov, A. and Jankovic, B. (2004). Application of 'inductive' QSAR descriptors for quantification of antibacterial activity of cationic polypeptides, *Molecules* **9**(12): 1034–1052.

Cortes, C. and Vapnik, V. (1995). Support-vector networks, *Machine Learning* **20**(3): 273–297.

Dudek, A. Z., Arodz, T. and Gálvez, J. (2006). Computational methods in developing quantitative structure-activity relationships (QSAR): a review., *Combinatorial chemistry & high throughput screening* **9**(3): 213–28.

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

Gilbert, D. N., Guidos, R. J., Boucher, H. W., Talbot, G. H., Spellberg, B., Edwards, J. E., Scheld, M., Bradley, J. S. and Barlett, J. G. (2010). The 10 x '20 Initiative: pursuing a global commitment to develop 10 new antibacterial drugs by 2020., *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America* **50**(8): 1081–3.

Golland, P., Liang, F., Mukherjee, S. and Panchenko, D. (2000). Permutation Test for Classification, *Journal of Machine Learning Research* **1**: 1–48.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning*, Vol. 18, second edn, Springer.

Hemmateenejad, B., Yousefinejad, S. and Mehdipour, A. R. (2011). Novel amino acids indices based on quantum topological molecular similarity and their application to QSAR study of peptides., *Amino acids* **40**(4): 1169–83.

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

Kiralj, R. and Ferreira, M. M. C. (2009). Basic validation procedures for regression models in QSAR and QSPR studies: Theory and application, *Journal of the Brazilian Chemical Society* **20**(4): 770–787.

Lin, Z. H., Long, H. X., Bo, Z., Wang, Y. Q. and Wu, Y. Z. (2008). New descriptors of amino acids and their application to peptide QSAR study.

Liu, D. C. and Nocedal, J. (1989). On the Limited Memory BFGS Method for Large Scale Optimization, *Mathematical Programming* **45**: 503–528.

Marr, A. K., Gooderham, W. J. and Hancock, R. E. (2006). Antibacterial peptides for therapeutic use: obstacles and realistic outlook., *Current opinion in pharmacology* **6**(5): 468–72.

Ng, A. (2009). Machine Learning.
URL: *https://www.coursera.org/learn/machine-learning/lecture/Kont7/learning-curves*

Ojala, M. and Garriga, G. C. (2010). Permutation Tests for Studying Classi er Performance, *Journal of Machine Learning Research* **11**: 1833–1863.

Pratim Roy, P., Paul, S., Mitra, I. and Roy, K. (2009). On two novel parameters for validation of predictive QSAR models., *Molecules (Basel, Switzerland)* **14**: 1660–1701.

Rao, H. B., Zhu, F., Yang, G. B., Li, Z. R. and Chen, Y. Z. (2011). Update of PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence, *Nucleic Acids Research* **39**(SUPPL. 2): 385–390.

Ruan, J., Wang, K., Yang, J., Kurgan, L. a. and Cios, K. (2005). Highly accurate and consistent method for prediction of helix and strand content from primary protein sequences, *Artificial Intelligence in Medicine* **35**(1-2): 19–35.

Scior, T., Medina-Franco, J. L., Do, Q.-T., Martínez-Mayorga, K., Yunes Rojas, J. a. and Bernard, P. (2009). How to recognize and workaround pitfalls in QSAR studies: a critical review., *Current medicinal chemistry* **16**(32): 4297–313.

Shin, H. C., Orton, M. R., Collins, D. J., Doran, S. J. and Leach, M. O. (2013). Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8): 1930–1943.

Shu, M., Yu, R., Zhang, Y., Wang, J., Yang, L., Wang, L. and Lin, Z. (n.d.). Predicting the Activity of Antimicrobial Peptides with Amino Acid Topological Information, *Medicinal Chemistry* (1): 32–44.

Smola, A. J. and Scholkopf, B. (2004). A tutorial on support vector regression, *Statistics and Computing* **14**(3): 199–222.

Taboureau, O. (2010). Methods for Building Quantitative Structure–Activity Relationship (QSAR) Descriptors and Predictive Models for Computer-Aided Design of Antimicrobial Peptides, *in* A. Giuliani and A. C. Rinaldi (eds), *Antimicrobial Peptides, Methods in Molecular Biology*, Vol. 8 of *Methods in Molecular Biology*, Humana Press, Totowa, NJ, chapter 6, pp. 77–86.

Todeschini, R. and Consonni, V. (2000). *Handbook of Molecular Descriptors*, Wiley-VCH, Federal Republic of Germany.

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

Tropsha, A. (2010). Best Practices for QSAR Model Development, Validation, and Exploitation, *Molecular Informatics* **29**(6-7): 476–488.

Wang, P., Hu, L., Liu, G., Jiang, N., Chen, X., Xu, J., Zheng, W., Li, L., Tan, M., Chen, Z., Song, H., Cai, Y. D. and Chou, K. C. (2011). Prediction of antimicrobial peptides based on sequence alignment and feature selection methods, *PLoS ONE* **6**(4): e18476.

Wang, Y., Ding, Y., Wen, H., Lin, Y., Hu, Y., Zhang, Y., Xia, Q. and Lin, Z. (2012). QSAR modeling and design of cationic antimicrobial peptides based on structural properties of amino acids., *Combinatorial chemistry & high throughput screening* **15**(4): 347–53.

Zhou, X., Li, Z., Dai, Z. and Zou, X. (2010). QSAR modeling of peptide biological activity by coupling support vector machine with particle swarm optimization algorithm and genetic algorithm, *Journal of Molecular Graphics and Modelling* **29**(2): 188–196.

Anexos

A. Clasificación de la actividad de péptidos

El diseño *in silico* de medicamentos ha sido usado en la predición y diseño de péptidos antibacterianos, que son considerados como una alternativa promisoria a los medicamentos actuales. Éstos presentan rápida acción y baja probabilidad de generar resistencia. Sin embargo, el espacio de búsqueda de péptidos candidatos a ser sintetizados es grande. Por esta razón, es necesario usar métodos computacionales que permitan reducir el número de secuencias a sintetizar. Para ello, se propuso el uso de Relaciones Cuantitativas entre Estructura-Actividad (QSAR, por sus siglas en inglés) junto con Máquinas de soporte vectorial para identificar péptidos antibacterianos sintéticos contra patógenos como *E. coli* y con Concentración Mínima Inhibitoria (CMI) menor a 10 uM.

Para realizar este proceso, se seleccionaron los péptidos y los descriptores que se iban a usar en la metodología QSAR. Inicialmente, se tomó un conjunto de 200 secuencias de péptidos junto con su respectivo CMI_{50} y se codificó su información estructural usando el punto isoeléctrico, hidrofobicidad, tamaño del péptido, tendencia de la estructura a forma α-helice, hoja β y turn, tendencia a formar agregados *invitro* e *invivo*. Adicionalmente, se ajustó los parámetros libres de la MSV (de la misma manera que la SVR, sólo que en este caso se tiene que optimizar el parámetro C y γ) junto con la validación cruzada. Con este esquema, se clasificó estos péptidos antibacterianos con una precisión del 88.47 % y un coeficiente de correlación de 0.762. Adicionalmente, se llevó a cabo el mismo procedimiento usando la base de datos DADP (Database of Anuran Defense Peptides), de la cual se extrajeron 479 secuencias de péptidos junto con su CMI. Para estos péptidos, el MIC fue reportado en los rangos de 0.3uM to 231uM. También se codificó la información estructural de cada secuencia y se desarrolló una MSV con los parámetros libres ajustados. El modelo de clasificación reportado mostró una precisión estimada del 72.73 % con un coeficiente

de correlación de 0.454. Estos resultados indican que usando QSAR y MSV es posible identificar secuencias de péptidos activos y con CMIs menores a 10 uM.