Predicción de propiedades farmacológicas de moléculas para el virus Chikungunya mediante el uso de machine leargning

Universitat Oberta de Catalunya

Miguel Jiménez Morcuende

MU Bioinf. i Bioest.
Drug Design and Structural Biology

Jorge Valencia Delgadillo

Nuria Pérez Álvarez

15/01/2023

UNIVERSITAT DE BARCELONA

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Predicción de propiedades farmacológicas de moléculas para el virus Chikungunya mediante el uso de machine leargning*
Nombre del autor:	*Miguel Jiménez Morcuende*
Nombre del consultor/a:	*Jorge Valencia Delgadillo*
Nombre del PRA:	*Nuria Pérez Álvarez*
Fecha de entrega (mm/aaaa):	*01/2023*
Titulación o programa:	*MU Bioinf. i Bioest.*
Área del Trabajo Final:	*Drug Design and Structural biology*
Idioma del trabajo:	*Castellano*
Palabras clave	*Chikungunya, Machine learning, Treatment*

Resumen del Trabajo

El Chikungunya (CHIKV) es un virus cuyos síntomas son fiebre alta y dolores intensos, llegando a producir problemas reumatológicos y neurológicos entre el 5% y el 50% de los afectados (Foeller et al., 2021). Entre 2004 y 2019 más de 10 millones de personas contrajeron la enfermedad, en 100 países diferentes (Suhrbier, 2019). La falta de una vacuna o medicamento recalca el peligro que supone esta enfermedad para la sanidad.

En este estudio se ha desarrollado un método de identificación de moléculas anti-CHIKV utilizando machine learning. Se recuperaron moléculas experimentalmente validadas de la base de datos ChEMBL con actividad EC50 frente al virus. Se seleccionaron una serie de descriptores mediante RDkit y se filtraron en función de la correlación y la variabilidad. Los datos fueron divididos en set de datos de entrenamiento y testeo. Diferentes técnicas de machine learning fueron implementadas como support vector machine, random forest, k- nearest neighbor, naive bayes, decision tree y extreme gradient boosting para construir los modelos de predicción.

Se han logrado valores de precisión de entre 0.644 a 0.722 y valores AUC de 0.667 a 0.786 en los modelos mejor desarrollados para los datos test.

De este modo se realizó un primer paso para la identificación de potenciales moléculas no utilizadas previamente como medicamento con efecto inhibidor frente a CHIKV.

Abstract

Chikungunya (CHIKV) is a virus whose symptoms are high fever and intense pain, causing rheumatological and neurological problems in between 5% and 50% of those affected (Foeller et al., 2021). Between 2004 and 2019, more than 10 million people contracted the disease in 100 different countries (Suhrbier, 2019). The lack of a vaccine or drug highlights the importance of this disease as public health concern.

In this study, a method for the identification of anti-CHIKV molecules has been developed using machine learning. Experimentally validated molecules with EC50 activity against the virus were retrieved from the ChEMBL database. A series of descriptors were selected using RDkit and filtered based on correlation and variability. The data was divided into training and testing data sets. Different machine learning techniques such as support vector machine, random forest, k-next nearn, naive bayes, decision tree and extreme gradient boosting were implemented to build the prediction models.

Precision values of between 0.644 to 0.722 and AUC values of 0.667 to 0.786 have been achieved in the best developed models for the test data.

So, a first step was carried out for the identification of effective molecules not previously used as drugs with an inhibitory effect against CHIKV.

Índice

Lista De Figuras

1. Introducción

1.1. Contexto y justificación del Trabajo

El virus chikungunya fue aislado por primera vez a mediados del sigo XX en Tanzania (Ross RW, 1956). La primera descripción de los síntomas de la enfermedad fue fiebre alta (al rededor de 38,3 y 39,4, aunque en muchos casos mayor), dolores punzantes que paralizaban al paciente, de forma ocasional se produjo sufusión de los ojos seguida de una conjuntivitis persistente y la irrupción de sarpullidos en las zonas de extensión y contracción de las extremidades (Matusali et al., 2019; Robinson, 1955). Otros síntomas característicos son la fatiga intensa, anorexia, nauseas, vómitos y diarrea, durando entre 7 a 10 días (Simon et al., 2011). Presenta síntomas muy similares a los del virus del dengue, pero con mayor presencia del artralgias (Erin Staples et al., 2009). De hecho, antes de su descubrimiento, muchos de los casos se diagnosticaban como malaria o dengue (Vairo et al., 2019). Al contrario que el ZIKV o el DENV, CHIKV causa síntomas en la mayoría de gente infectada, oscilando entre el 72% y el 95% de estos (Matusali et al., 2019).

Se trata de un alfavirus que consiste en un ARN monocatenario positivo y una estructura envuelta (Burt et al., 2017). El genoma (Figura 1) es de aproximadamente 11,8 kb, con dos marcos de lectura abiertos (ORF), el extremo 5’ que se encarga de codificar las 4 proteínas no estructurales (nsP1-nsP4) y el extremo 3’ que codifica la estructuras proteicas (C, E1-E3 glicoproteínas y 6K) (da Silva-Júnior et al., 2017; Galán-Huerta et al., 2015). La heterogeneidad del genoma está relacionada con la falta de corrección de errores durante la síntesis de ARN, lo que conlleva mutaciones y, por tanto, la aparación de numerosas cepas (Higuera & Ramírez, 2019).
Nsp1 participa en la replicación viral debido a sus múltiples actividades enzimáticas como la actividad ATPasa y trifosfatasa ARN, helicasa de ARN y actividad proteasa (Utt et al., 2015). Nsp2 actúa neutralizando la respuesta antiviral del hospedador, interfiriendo en la vía de señalización JAK/STAT e induciendo la autofagia (Fros et al., 2010). Nsp3 promueve interacciones con numerosas proteínas celulares y es capaz de unirse a polimeros cargados negativamente (Panas et al., 2014). Nsp4 actúa como una ARN polimerasa dependiente y cataliza la formación de ARN de sentido negativo (Chen et al., 2017). C es la cápside del virus y las glicoproteínas E1, E2 y E3 forman el complejo de replicación viral para la síntesis del genoma. Todavía no se sabe con certeza, pero la proteína 6K parece estar relacionada con el ensamblaje y estados de unión de la superficie del virón a la célula infectada (Moizéis et al. 2018).

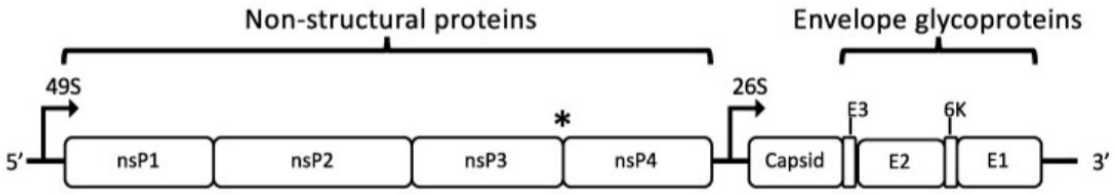

Figura 1. *Representación esquemática del genoma de un alfavirus (Bakar & Ng, 2018).*

El vector de transmisión del virus son los mosquitos hembra de la família *Aedes* (*aegypti* y *albopictus)* que migran por zonas tropicales del este/centro y sur de África, sudeste asiático y sudamérica (Khongwichit et al., 2021; Silva et al., 2018), donde se describen tres linajes enzoóticos, dos de ellos en el continente Africano (el Occidental y Este, Centro y Sur de África) y el asiático (Lo Presti et al., 2012). Silva et al., en 2018, aceptaron una nueva variante denominada Océano Índico.
Transmiten el virus en humanos (transmisión horitzontal) a través de su picadura, ya que inyectan su saliva, la cual puede contener patógenos, al interior del anfitrión (Tolle, 2009), que infectan fibroblastos, queratinocitos y macrófagos residentes (Schwartz & Albert, 2010). Finalmente, el virus se extiende por órganos, músculo esquelético, órganos linfoides, las articulaciones y el sistema nervioso central (Valdés López et al., 2019).

Esta amenaza ha dejado de afectar no solo a estas regiones, sino que se está extendiendo por zonas con climas más templados como el sur de Europa, así como Centroamérica o el sur de Estados Unidos, debido al aumento de las temperaturas (Enserink, 2007), a un mayor número de desastres naturales como inundaciones (depositan sus huevos directamente en el agua o en superficies húmedas [White, 2004]) (Rezza et al., 2007), o la globalización (Bhatia & Narain, 2009). A este riesgo se le suma el hecho de que también existe la transmisión entre madre y feto, pero solo cuando esta es infectada los días previos al parto (Caglioti et al., 2013).

En 2011 se estimaron entre 33.000 y 93.000 casos clínicos al año de Chikungunya, donde se estimó que el 39% de la población mundial vivía en países endémicos para CHIKV y se encuentran en riesgo de infección (LaBeaud et al., 2011). Se cree que entre 2004 y 2019 la enfermedad ha llegado a más de 100 países (Figura 2), produciendo más de 10 millones de casos y poniendo en riesgo de infección a 1,3 billones de personas (Suhrbier, 2019).

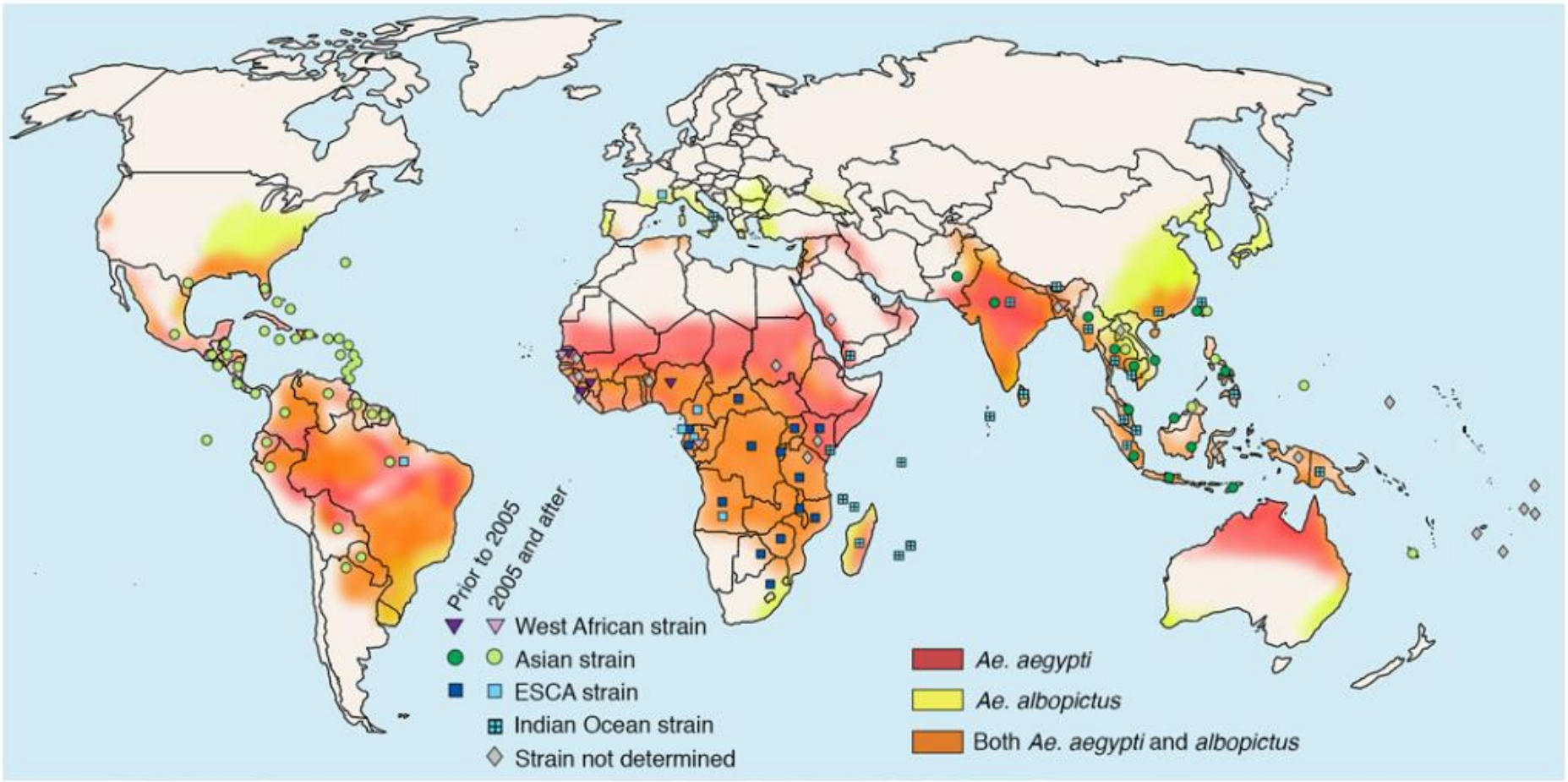

Figura 2. *Distribución geográfica de CHIKV endémico y de sus vectores primarios Ae. Aegypti y Ae. Albopictus (Silva et al., 2018). En rojo se encuentra el área de distribución del hábitat de Ae. Aegypti, en amarillo el de Ae.albopictus y en naranja donde conviven ambos mosquitos. En cuanto a las variantes, en forma de triángulo se encuentra la variante Africana Occidental, en forma de círculo la variante asiática. Como un cuadrado la segunda variante africana proveniente del este, centro y sur de África. Con cuadaros con una cruz la variante océano índico y con un rombo, aquellas cepas que no han sido determinadas.*

Se trata de una enfermedad que no tiene cura de la cual solo se puede tratar sus síntomas para aliviarlos. La mortandad es baja produciéndose principalmente en pacientes con patologías previas (Economopoulou et al., 2009), pero entre el 5% al 50% de los supervivientes presentan complicaciones reumatológicas (artritis y poliartralgias) y neurológicas (depresión y fatiga) crónicas (Foeller et al., 2021; Soumahoro et al., 2009).

Por consiguiente, se presenta un tema muy interesante cuya investigación ha incrementado drásticamente en la última década debido al riesgo de apariciones masivas de esta enfermedad, gracias a las mejoras de las condiciones para su expansión y el alto grado de exposición a gente sin contacto previo. Esto se ve reflejado en la base de datos pubMed (https://pubmed.ncbi.nlm.nih.gov) la cual registra un total de 6.999 artículos relacionados con Chikungunya, dónde 5.463 de ellos han sido publicados entre el 2012 y el 2022.

Es un hecho irrefutable que existe un gran interés por parte de la comunidad científica de informar sobre este virus y, por tanto, de encontrar una cura a esta enfermedad cada vez más presente en “países desarrollados”, debido al riesgo de ser una epidemia y provocar colapsos en los sistemas sanitarios.

En consecuencia, mediante análisis de machine learning, se quiere predecir propiedades farmacológicas de biomoléculas que produzcan actividad sobre el virus chikungunya.

En este Trabajo se creará una serie de modelos que ayuden a la predicción de nuevas moléculas que presenten actividad para la enfermedad del virus chikungunya, mediante un número de descriptores obtenidos a partir de un conjunto de éstas, de actividad conocida para Chikungunya.

1.2. Objetivos del Trabajo

Objetivos generales:

1. Creación de una serie de modelos que ayuden a la predicción de nuevas moléculas con actividad sobre esta enfermedad en concreto.

2. Comparativa de los modelos creados y selección del más fiable para futuros usos.

Objetivos específicos

1.1. Obtención del set de datos de moléculas que presentan actividad frente a el virus del chikungunya y cálculo de los descriptores

1.2. Identificación de las variables más descriptivas para su empleo en los entrenamientos.

1.3. Entrenamiento de los diferentes modelos.

2.1. Realización de curvas ROC y análisis de rendimiento de los diferentes resultados obtenidos para cada modelo.

2.2. Optimización del modelo seleccionado.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

El trabajo no presenta ningún impacto a nivel de sostenibilidad puesto que no presenta ningún cambio para el medioambiente ni en la huella ecológica, por lo que no supone ninguna modificación en el clima ni en la vida. La aparición masiva de estos mosquitos está relacionada con zonas de aguas estancadas y aunque se busque una cura para el virus siempre será necesario el control de estas poblaciones mediante pesticidas.

Tampoco presenta impacto del tipo ético-social ya que no supone ningún cambio en los aspectos éticos y sociales, así como en el marco normativo/legislativo, ni en las condiciones de vida de la población.

Para acabar, en la dimensión de diversidad, género y derechos humanos, sí que supone un impacto positivo puesto que, como se ha explicado con anterioridad, esta enfermedad supone un gran riesgo en países en vías de desarrollo. En estas zonas, se encuentran el hábitat de los mosquitos del género *Aedes* lo que supone una mayor exposición y riesgo a la transmisión. Además, las condiciones de vida y sanidad son inferiores que, a otros países como los europeos o Estados Unidos, por lo que sus sistemas de sanidad son más probables a colapsar frente a este tipo de epidemias. La falta de materiales de prevención, también facilitan la propagación y que se produzcan un mayor número de mutaciones. La ayuda de este trabajo a la creación de un medicamento frente al virus podría reducir la mortandad y evitar un posible colapso de los servicios públicos, puesto que el medicamento iría dirigido de forma principal a esta población más vulnerable.

No solo tendría impacto a nivel de la salud sino también a niveles económicos, porque muchos de estos países viven del turismo. Un ejemplo muy claro fue el brote que surgió en las islas La Réunion, al este de Magadascar, que supuso unas pérdidas de 160 millones de dólares (Enserink, 2007).

1.4. Enfoque y método seguido

En el desarrollo novel de un medicamento, el valor total de gastos puede ser aproximadamente de 2.558 mil millones de dólares, durante un periodo largo de años donde tan sólo el 13% de ellos tienen éxito (DiMasi et al., 2016).

Mediante el uso de técnicas informáticas se consigue proveer no sólo las propiedades moleculares, sino que también proporciona los atributos ideales *in silico* junto con una gran reducción de los costes preclínicos (Wale, 2011).

Las técnicas de investigación mediante estructuras de proteínas 3D también requieren tiempo y dinero. Mediante machine learning solo es necesario conocer la estructura secundaria y las interacciones entre proteínas residentes. (Dara et al., 2022)

También evita la necesidad de realizar pruebas con animales, la cuales han recibido una publicidad negativa en los últimos tiempos (Elbadawi et al., 2021)

Por tanto, se emplearán las técnicas de machine learning para obtener unos modelos que ayuden a la predicción de nuevas moléculas con actividad para esta enfermedad.

En un experimento ideal, se utilizaría como diana proteínas que formen parte del virus (Gupta et al., 2021) como la glicoproteína E2 que es una transmembrana de tipo I relacionada con la unión del virus a la membrana del anfitrión (Delogu et al., 2011a) o la proteasa nsP4 que es una polimerasa dependiente de ARN altamente conservada (Bakar & Ng, 2018), pero por falta de tiempo y por que todavía no se conocen para algunas proteínas los mecanismos de acción exactos, se ha empleado la molécula completa del virus (Battisti et al., 2021). Además, al aplicar la búsqueda a todas las moléculas con actividad para Chikungunya, se asegura una mayor base de datos con la que trabajar y, por tanto, tener más ejemplos de entrenamiento (mayor fiabilidad).

Se seleccionaron biomoléculas activas que presentan actividad para el virus chikungunya con un EC50 igual o menor al valor de corte 7000nM, teniendo en

cuenta sus 129 variables descriptoras para la predicción de estas nuevas moléculas. Se realizó una limpieza de los datos para aquellos que estaban repetidos o presentaban poca variabilidad o correlación entre sí.

Los modelos se realizaron mediante RMarkdown utilizando la versión de Rstudio 2022.07.2 + 576 y se emplearon las siguientes librerías para el entrenamiento de los modelos: Naive Bayes ("e1071"), Support Vector Machine ("kernlab"), Random Forest ("randomForest"), k – Nearest Neighbor ("class"), Decision trees ("C5.0") y Extreme Gradient Boosting ("xgboost")

Una vez obtenidos los modelos se ha pasado a realizar un análisis de los resultados teniendo en cuenta factores como el número de falsos positivos, de falsos negativos, verdaderos positivos, verdaderos negativos, la tasa de error, la precisión, valor kappa, la especificidad, la sensibilidad y el valor AUC mediante las curvas ROC (librería ROCR y pROC).

Finalmente, del modelo seleccionado se ha producido una modificación de sus hiperparámetros para mejorar el modelo y así sea más preciso en sus predicciones para otras moléculas.

Para realizar el TFM, se empleó un ordenador HP OMEN 25L GT15-0011ns, Intel® Core™ i7-12700F, 16GB RAM, 1 TB SSD, NVIDIA® GeForce RTX™ 3060. No se prevén costes económicos extra asociados al producto.

1.5. Planificación del Trabajo

Cada día contemplado en el calendario tendrá una dedicación aproximada de 4 horas (sin contar fines de semana), con tal de llegar al mínimo de 300 horas de dedicación al TFM. Esto no significa que el tiempo aplicado haya sido distribuido siempre de una forma constante.

Tareas

Las tareas realizadas para desempeñar el trabajo final de máster son:

T.1. Selección de una enfermedad para la cual buscar biomoléculas que presenten actividad y sean suficientes para crear una base de datos para el entrenamiento.

T.2. Revisión bibliográfica sobre la enfermedad y como se encuentra su investigación en la actualidad.

T.3. Obtención del set de datos tomando moléculas con un EC50 igual o menor a 7000nM.

T.4. Generación de los descriptores mediante RDKit

T.5. Procesamiento de los datos. Limpieza de aquellos datos que pudieran estar repetidos y selección de los descriptores eliminando aquellas variables con una correlación de Pearson de $\pm$ 0.8 y aquellas que muestren poca varianza.

T.6. Entrenamiento de los diferentes modelos (K-Nearest Neighbor [k-NN], Support Vector Machine [SVM], Naive Bayes [NB], Decision trees [DT], Extreme Gradient Boosting [XGB] y Random Forest [RF]).

T.7. Análisis de los resultados obtenidos, realización de las curvas ROC y selección del modelo con mejor función predictora.

T.8. Optimización de hiperparámetros para el modelo seleccionado.

T.9. Extracción de conclusiones a partir de los resultados y posible implicación como herramienta para la predicción de moléculas con actividad sobre CHIKV.

Calendario

La planificación llevada a lo largo del TFM se puede observar en la Tabla 1.

***Tabla 1**. Planificación del TFM.*

Tareas	Inicio	Entrega
PEC1- Definición y plan de trabajo	27/09/2022	17/10/2022
T.1. Selección de una enfermedad	27/09/2022	06/10/2022
T.2. Revisión bibliográfica	06/10/2022	17/10/2022
T.3. Set de datos	06/10/2022	12/10/2022
T.4. Generación de descriptores	10/10/2022	16/10/2022
PEC2- Desarrollo del trabajo - fase I	18/10/2022	21/11/2022
T.4. Revisión del set de descriptores	18/10/2022	26/10/2022
T.5. Procesamiento de los datos	27/10/2022	4/11/2022
T.6. Entrenamiento de los modelos	5/11/ 2022	21/11/2022
PEC3- Desarrollo del trabajo – fase II	22/11/2022	24/12/2022
T.6. Entrenamiento de los modelos	22/11/2022	13/12/2022
T.7. Análisis de los resultados y curvas ROC	14/12/2022	24/12/2022
T.8. Optimización de hiperparámetros	24/12/2022	28/12/2022
T.9. Conclusiones	28/12/2022	02/01/2023
PEC4-Cierre de la memoria y de la presentación	27/12/2022	15/01/2023
Redacción memoria	28/12/2022	12/01/2023
Elaboración presentación	12/01/2023	15/01/2023
Defensa pública	23/01/2023	03/02/2023

En el siguiente diagrama de Gantt (Figura 3), elaborado mediante “Team Gantt” (https://www.teamgantt.com), se puede ver la planificación de la distribución de las tareas en el tiempo:

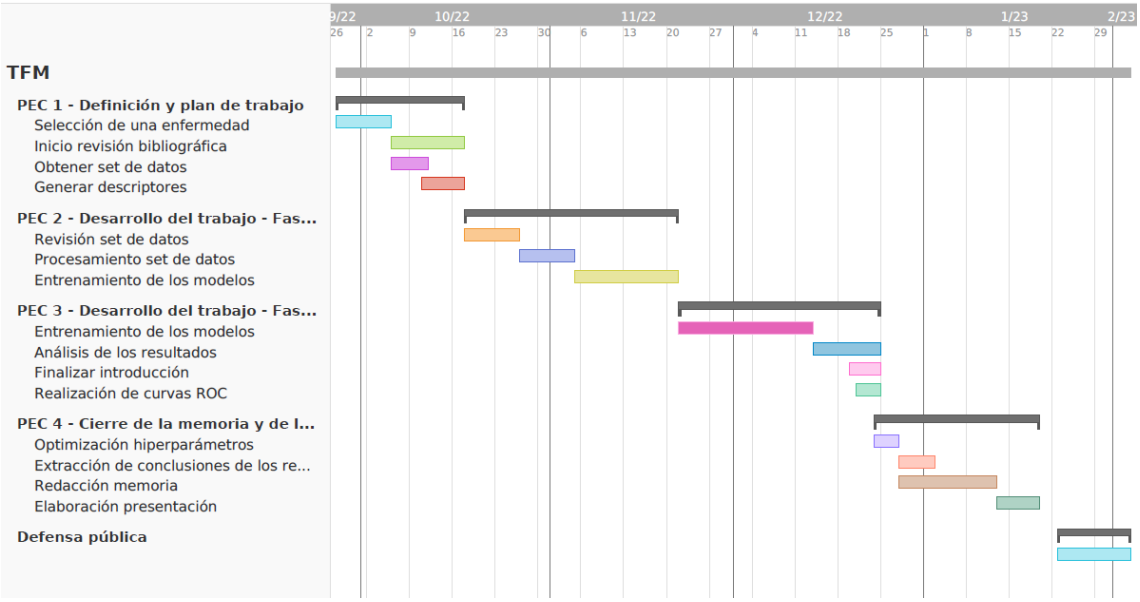

***Figura 3**. Diagrama de Grantt.*

Hitos

Los hitos designados por el Plan Docente de la asignatura del TFM son:

***Tabla 2.** Hitos del TFM*

PEC1- Definición y plan de trabajo	27/09/2022	17/10/2022
PEC2- Desarrollo del trabajo - fase I	18/10/2022	21/11/2022
PEC3- Desarrollo del trabajo – fase II	22/11/2022	24/12/2022
PEC4- Cierre de la memoria y de la presentación	28/12/2022	15/01/2023
Defensa pública	23/01/2023	03/02/2023

Descripción de los hitos de la Tabla 2:

PEC1 - Plan de trabajo: Se establece con que enfermedad se va a estudiar, con una breve explicación sobre ésta, cuáles son los objetivos que se buscan en este TFM y los tiempos de realización de las tareas para cumplir esos objetivos.

PEC2 - Desarrollo del trabajo – Fase I: Realización de la parte técnica del proyecto con la obtención del set de datos, la generación de descriptores, el procesamiento de los datos y el inicio de los entrenamientos de los modelos.

PEC3 – Desarrollo del trabajo – Fase II: Continuar los entrenamientos con los modelos y analizar los resultados obtenidos para determinar cuál es el modelo con mejor capacidad de predicción.

PEC4 – Cierre de la memoria y de la presentación: Se optimiza los hiperparámetros del modelo seleccionado y se finaliza la memoria con unas conclusiones y se realiza la presentación del TFM.

Defensa pública del TFM.

Análisis de riesgos

Los riesgos que pueden surgir a lo largo del desarrollo de este trabajo son:

- Que los resultados obtenidos de los modelos de entrenamiento tengan unos valores de confianza inferiores a lo esperado y, por tanto, no obtener unos modelos que ayuden correctamente a la predicción de nuevas moléculas frente a la enfermedad.

- Que no se observen diferencias aparentes entre los modelos empleados y no se puede hacer una criba o selección de los mejores métodos, por la falta de tiempo para realizar posibles optimizaciones en todos los modelos empleados.

1.6. Breve sumario de productos obtenidos

El producto que se generará con la realización de este TFM será un modelo de predicción de actividad frente al chikungunya virus a partir de una base de datos previamente elaborada, para determinar nuevas moléculas que presenten actividad frente a este virus y, por tanto, su posible desarrollo como fármaco para esta enfermedad, en caso de la existencia de actividad.

1.7. Breve descripción de los otros capítulos de la memoria

En el apartado del Estado del arte se describirá la hipótesis de partida de este TFM y como se plantea resolverla, teniendo en cuenta la actualidad científica en referencia a la investigación de esta enfermedad y los métodos de machine

learning empleados por grupos de investigación en otras enfermedades para la creación de los modelos de predicción.

En Material y métodos se hará una descripción de la obtención del set de datos, cálculo de descriptores, la preparación de los datos, modelos empleados y eficacias.

En Resultados y Discusión se observarán los resultados obtenidos de cada modelo y sus parámetros, se valorarán y se discutirán cuáles son los modelos más representativos y su uso para la predicción de actividad en nuevas moléculas.

Por último, se harán unas conclusiones de los resultados obtenidos si son los esperados o difieren de otros trabajos, si se ha logrado alcanzar todos los objetivos propuestos, así como líneas de trabajo futuro.

2. Estado del arte

Actualmente no existe ningún tipo de vacuna ni medicamento frente al Chikungunya (Mourad et al., 2022), y el único tratamiento que existe es el alivio de los síntomas con antipiréticos, corticoides esteroideos, analgésicos y antinflamatorios no esteroideos (NSAIDs) (Abdelnabi et al., 2017) , pero se está siguiendo un desarrollo activo para su diseño.

En relación a las vacunas, existen hasta 10 estudios que se encuentran actualmente en fase III de ensayo clínico (https://www.clinicaltrials.gov), fase en la que se verifican de manera extenuada los aspectos de seguridad y eficacia del fármaco (https://www.aemps.gob.es). La vacuna en estudio VLA1553 obtuvo niveles de seroprotección en el 98.9% de los participantes después de un mes de la vacunación y en el 96.3% después de 6 meses (https://valneva.com). Además, solo presentó en un 0.45% de los participantes efectos secundarios adversos importantes (https://www.clinicaltrials.gov).

En cuanto al diseño de fármacos, se está realizando una identificación de los inhibidores de CHIKV, siguiendo 4 distintos tipos de enfoque (Pérez-Pérez et al., 2019):

- Reutilización de medicamentos aprobados/existentes: se busca una nueva indicación para este medicamento o para explorar nuevos objetivos moleculares o vías reduciendo los tiempos de coste y desarrollo (Powers, 2018).

- Detección basada en células fenotípicas: identificación de compuestos que interfieran en la replicación de actividad desconocida, pudiendo ser naturales o artificiales los cuales pueden ser medidos mediante métodos colorimétricos o por luminiscencia (Tsao et al., 2018).

- Detección basada en objetivos: el ensayo se encuentra especialmente diseñado para identificar componentes que interfieran en un proceso particular que es necesario para la replicación del virus, tanto en ensayos *in vitro* como en ensayos *in cellulo* (Coles et al., 2003).

- Diseño basado en estructuras: identificación de nuevos agentes contra proteínas del virus mediante detección virtual de bases de datos químicas utilizando estructura de rayos x o modelos de homología (Malet et al., 2009).

Existe una serie de compuestos con posibilidad de ser antivirales frente al CHIKV. A continuación, se hará una breve descripción de ellos en función a su mecanismo.

Inhibidores de entrada (Figura 4):

- Arbidol: medicamento desarrollado para la profilaxis y tratamientos frente a infecciones respiratorias agudas, afectando a la glicoproteína transmembrana de tipo I E2, relacionada con la unión a la membrana del anfitrión (Delogu et al., 2011b).

- Cloroquina: medicamento diseñado contra la malaria, que afecta al pH del anfitrión (Khan et al., 2010).

- Epigalocatequina galato: extracto de té verde con capacidad antiviral contra VIH o hepatitis C entre otros (Weber et al., 2015).

- EIPA: 5-(N-etil-N-isopropil) -amilorida: molécula sintética que bloquea el intercambio de Na(+)/H(+), modificando el pH (Hua et al., 2019).

- Fenotiazina: medicamento antipsicótico, diseñado para reducir las alucinaciones y los delirios asociados con la psicosis (https://www.drugs.com), se desconoce el mecanismo de acción, pero evitó la replicación del virus (Pohjala et al., 2011).

- Harringtonina: se trata de un extracto natural de *Cephalotaxus harringtonia,* donde se observó una bajada en la concentración de las proteínas nsP3 y E2, inhibiendo la síntesis de estas proteínas, lo que produce un descenso de la virilidad (Kaur et al., 2013).

- IIc: análogo del arbidol, actúa sobre la proteína E2, aunque se cree que puede presentar otro mecanismo de acción (di Mola et al., 2014).

- Micafungina: tratamiento frente a la candidiasis, presenta relación las glicoproteínas de la envuelta (E1 y E2) (Ho et al., 2018).

- Niclosamida y nitazoxanida: inhibición en la transmisión célula a célula una de las principales vías de transmisión que permite a virus evitar el sistema inmune (Wang et al., 2016).

- Obatoclax: inhibe la fusión viral de CHIKV debido a su acción sobre E1 (Varghese et al., 2017).

- Rodanina: interactúa con la proteasa nsP2, mostrando interacciones hidrofóbicas con 3 residuos aminoacídicos (TYR1047, TYR1049 y TRP1084) en el bolsillo S3 de la proteína (Jadav et al., 2015).

- Tiazolidinediona: mecanismo similar al de la rodanina, solo que esta interactúa con los residuos CYS1013, TYR1047, TYR1049 y TRP1084) del bolsillo S2 de la proteasa nsP2 (Jadav et al., 2015).

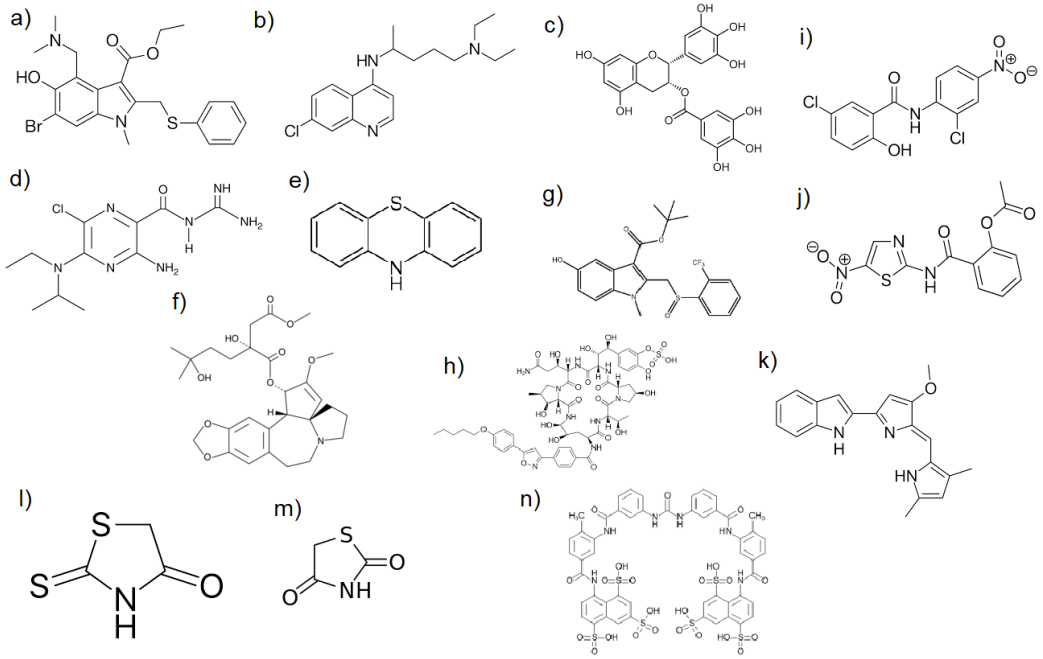

***Figura 4**. Estructura química de agentes inhibidores de entrada: a) Arbidol, b) Cloroquina, c) Epigalocatequina galato, d) 5-(N-etil-N-isopropil) -amilorida (EIPA), e) Fenotiazina, f) Harringtonina, g) IIc, h) Micafungina, i) Niclosamida, j) Nitazoxanida, k) Obatoclax, l) Rodanina, m) Tiazolidinediona, n) Suramina*

- Suramina: compuesto aprobado contra la tripanosomiasis inhibe los primeros ciclos de la replicación de CHIKV, impidiendo la unión viral a la membrana del anfitrión debido a inhibir cambios conformacionales en la envuelta de glicoproteínas del virus, aunque presenta efectos secundarios a largo plazo (Albulescu et al., 2020).

Inhibidores de la proteína nsP1 no estructural (Figura 5):

- 6'-β-fluoro-homoaristeromicina: son análogos de adenosina carbocíclica, que inhiben la actividad MTasa (metiltransferasa) de nsP1 (Kovacikova & van Hemert, 2020).

- Ácido lobárico: compuesto natural, que interfiere en el paso de guanilación del proceso de *capping* de nsP1 (Feibelman et al., 2018).

- MADTP: produce la sustitución del aminoácido P34S en el dominio funcional GTasa (Delang et al., 2016).

- Quininas: medicamento frente a la malaria, inhibe *in vitro* al virus gracias a una mutación que produce en la proteína no estructural a altas concentraciones (de Lamballerie et al., 2009).

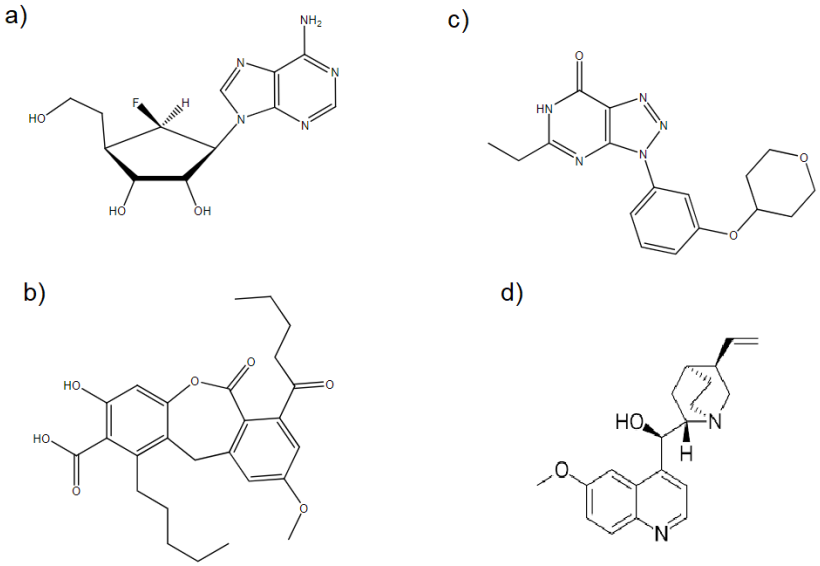

***Figura 5**. Inhibidores de la proteína nsP1: a) 6'-β-fluoro-homoaristeromicina, b) Ácido lobárico, c) MADTP, d) Quininas.*

Inhibidores de la proteína nsP2 no estructural (Figura 6):

- Peptidomimético (pequeñas moléculas no peptídicas que imitan la función de un péptido), Nelfinavir (medicamento frente al VIH), Telmisartan (mediacamento para la presión arterial alta) y thiazolidina-4-uno (derivado de la arilalquiladina) presentaron efectos inhibitorios frente a la protesasa nsP2 (Bhakat et al., 2015; Singh et al., 2018; Tripathi et al., 2020).

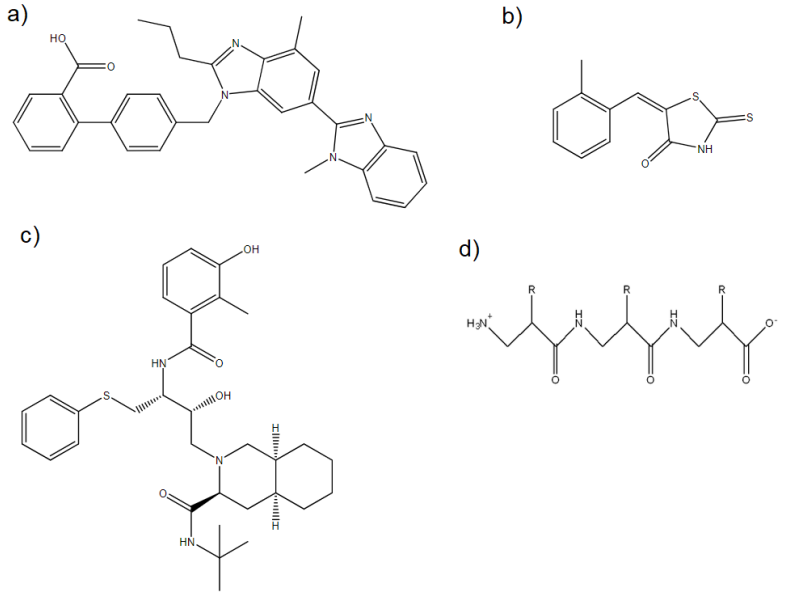

***Figura 6**. Inhibidores de la proteína nsP2: a) Telmisartan, b) Thiazolidina-4-uno, c) Nelfinavir, d) Peptidomimético.*

Inhibidores de la proteína nsP3 no estructural:

- En la actualidad, no se existe ningún tipo de molécula que haya presentado capacidad inhibitoria frente a CHIKV afectando a la proteína estructural nsP3, aunque se tienen listados de compuestos capaces de interactuar con esta proteína como los flavonoides actuando como posibles candidatos (Nguyen et al., 2014).

Inhibidores de la proteína nsP4 no estructural (Figura 7):

- Compuestos relacionados con benzimidazol: compuesto que actúa sobre el residuo M2295I que se encuentra en el dominio polimerasa ARN dependiente (RdRp) de nsP4 (Wada et al., 2017).

- Favipiravir: tratamiento antivírico contra diferentes tipos de virus ARN, previniendo el desarrollo de enfermedades neuronales severas y aumentando la ratio de supervivencia en ratones (Delang et al., 2014).

- NHC: β-D-N^4-hidroxicitidina mostró actividad en los primeros pasos de la replicación de CHIKV, indicando como objetivo potencial el dominio RdRp (Urakova et al., 2018).

- Sofosbuvir: medicamento frente al virus de la hepatitis C, protegió en ensayos de ratones la capacidad de infección del virus y aumentó la supervivencia (Santana et al., 2021).

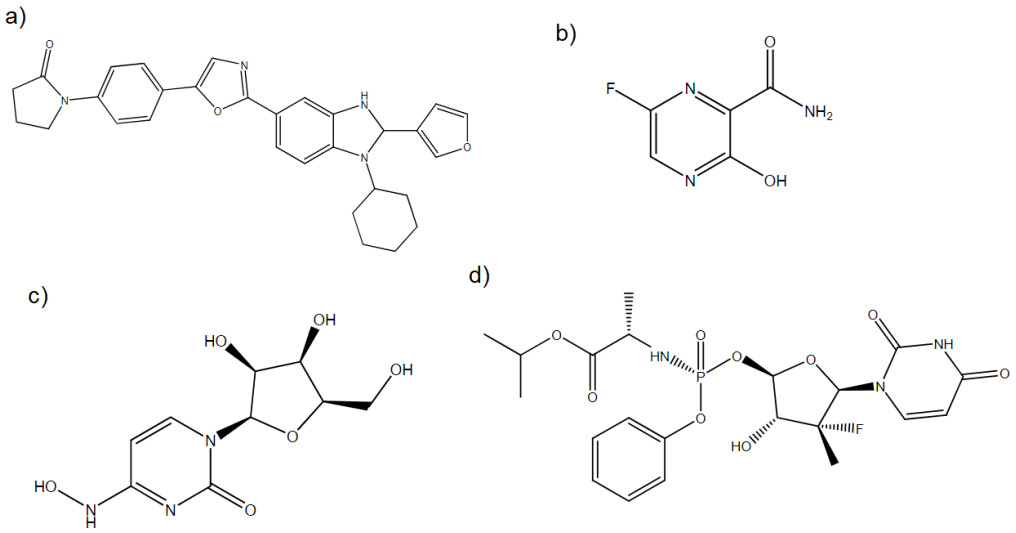

Figura 7*. Inhibidores de la proteína nsP4: a) Compuestos relacionados con benzimidazol, b) Favipiravir, c) β-D-N^4-hidroxicitidina (NHC), d) Sofosbuvir.*

Inhibidores de la proteasa de la cápside (C) (Figura 8):

- Ácido picolínico (PCA): es un quelante derivado de la piridina, que se une a la región hidrofóbica de la proteína de la cápside, interfiriendo en su interacción con el dominio citoplasmático de la glicoproteína E2 (Fernández et al., 2010).

- P1,P4-Di(adenosina-5') tetrafosfato (AP4), acetato de Eptifibatida (EAC) y sulfato de Paromomicina (PSU): los tres componentes redujeron la producción de virus infecciosos, principalmente en los estadios finales del ciclo viral cuando la producción de cápsides es máxima (Fatma et al., 2020).

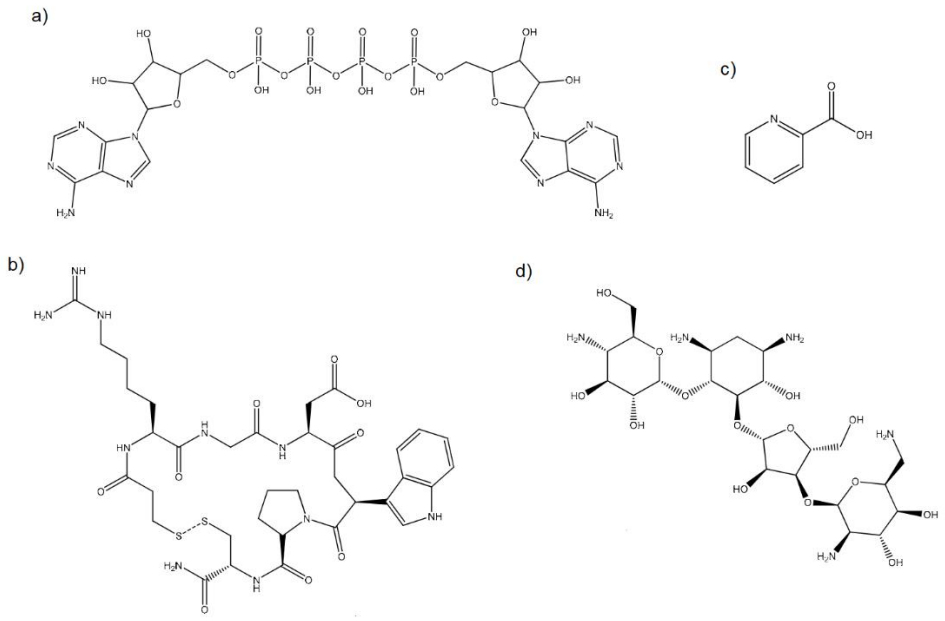

***Figura 8**. Inhibidores de la proteasa de la cápside (C): a) P1,P4-Di(adenosina-5') tetrafosfato (AP4), b) Acetato de Eptifibatida (EAC), c) Ácido picolínico (PCA), d) Sulfato de Paromomicina (PSU).*

Inhibidores de la proteína 6K (Figura 9):

- Amantadina: medicamento frente a la gripe, dificulta la actividad del canal iónico de CHIKV y altera la morfología de sus partículas víricas (Dey et al., 2019).

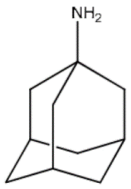

***Figura 9**. Inhibidor de la proteína 6K: a) Amantadina.*

También existe otra serie de agentes antivirales que actúan sobre factores y condiciones del anfitrión, teniendo un posible impacto sobre el ciclo de replicación viral como inhibidores de las vías lipídicas, inhibidores de la síntesis de purinas y pirimidinas, inhibidores de la síntesis de proteínas, inhibidores celulares proteicos, inhibidores de receptores celulares o agentes inmunomoduladores (todos los estudios han sido realizados sobre líneas celulares MRC5, BHK-21, Vero y Huh-7 a excepción de la suramina, el favipiravir y el sofosbuvir donde se han realizado ensayos en ratones líneas C57BL/6,

Ag129 y SwissWebster respectivamente) (Battisti et al., 2021; da Silva-Júnior et al., 2017) .

3. Materiales y Métodos

Esta sección describe la creación del set de datos y los métodos utilizados para mejorar los datos de entrenamiento mediante la selección correcta de descriptores. Para la predicción de actividad inhibitoria en las moléculas, se entrenó el set de datos y se comparó en diferentes modelos de machine learning, para comprobar su eficacia y posible uso futuro como mecanismo de selección de moléculas anti-CHIKV (Figura 10).

3.1. Set de Datos

De la base de datos ChEMBL ID (Gaulton et al., 2017), un total de 272 moléculas (tras borrar duplicados) cuyo objetivo es el ChEMBL ID 4296563, fueron seleccionadas. Las moléculas con EC50 igual o inferior a 7µM se consideraron como positivas. En total 129 moléculas no presentaban actividad mientras que 143 si que presentan actividad frente a la partícula del virus. Todos los inhibidores se consideraron como clase activa de valor 1 y los que no presentan actividad de valor 0.

Se calcularon un total de 129 descriptores mediante RDkit (2022.09.1) (Gawriljuk et al., 2021), incluyendo descriptores de la estructura general (Número de heteroátomos, hidrógenos aceptadores, hidrógenos donadores, enlaces amida, anillos aromáticos…), HallKierAlpha o descriptores tipo MOE (PEOE_VSA, SMR_VSA, SlogP_VSA…) entre otros. (Una mayor descripción del significado de los descriptores se encuentra en el anexo).

Se realizó una comprobación de los descriptores y en primer lugar se eliminaron aquellas variables con valor 0 para todas las moléculas, lo que significaba que la propiedad no estaba aportando ninguna información.

A continuación, se realizó un análisis del coeficiente de Pearson (Wang et al., 2013), eliminando los descriptores moleculares que presentasen un coeficiente de correlación de ±0.8 entre sí, puesto que se tratan de datos redundantes que no aportan más información.

También se tuvo en cuenta la poca varianza de los descriptores y aquellas columnas que presentasen valores constantes para las moléculas.

De los 129 descriptores, solo 63 descriptores se emplearon para el entrenamiento de los modelos (en el anexo se encuentra una tabla con todos los descriptores empleados).

Finalmente, se realizó la normalización de los datos mediante una función de normalización de máximos y mínimos.

Los datos se dividieron de manera aleatoria en training (0.67) y test (0.33) con una proporción de no activos y activos de 82 y 100 para los datos training y; 47 y 43 para los datos test.

3.2. Métodos de Machine Learning

Se han desarrollado algoritmos de predicción para los objetivos utilizando 6 técnicas de entrenamiento de datos: k-NN, NB, SVM, RF, DT y XGB. Los modelos han sido elegidos en función a trabajos previos que implementan el uso de estos modelos en el desarrollo de nuevos fármacos (Fang et al., 2013; Gawriljuk et al., 2021; Jiang et al., 2020; Kamboj et al., 2022; Matsumoto et al., 2016).

K-Nearest Neighbors

El algoritmo k-NN utiliza la información sobre los kvecinos más cercanos de un ejemplo para clasificar ejemplos sin etiquetar. La molécula es clasificada por la mayoría de los votos de sus vecinos, siendo la molécula asignada a la clase más común entre sus *k* vecinos más cercanos (Lavecchia, 2015).

Al tomar una *k*, el algoritmo requiere un conjunto de datos de entrenamiento compuesto por ejemplos que han sido clasificados previamente, etiquetados por una variable nominal.

Se trata de un método simple y efectivo, a la par que rápido, que no toma suposiciones de la distribución de los datos. En contra tiene que no produce un modelo como tal, por lo que se limita la capacidad para entender como los descriptores están relacionados con la variable a predecir, la selección de una *k* apropiada, una clasificación lenta y que las variables nominales y los datos perdidos requieren procesamiento adicional (Lantz, 2015).

La letra k es un término variable que implica que se puede utilizar cualquier número de vecinos más cercanos (Venables & Ripley, 2002).

Para la función knn() del paquete “class”, se le dio a *k* valores de 1, 5 y 11.

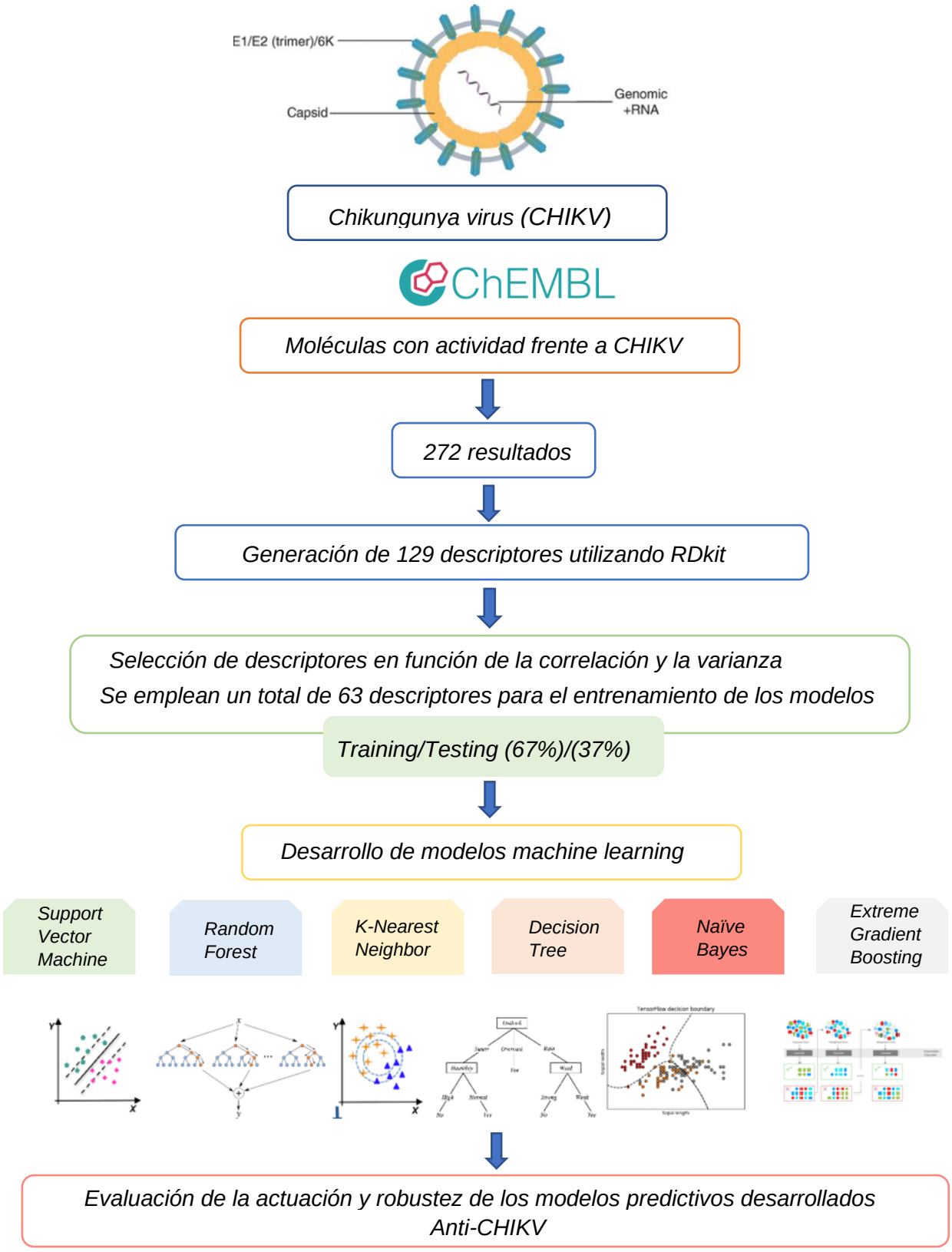

***Figura 10**. Esquema general de la metodología anti-CHIKV para el desarrollo de algoritmos predictivos para identificar inhibidores. Las moléculas fueron tomadas de ChEMBL. Los descriptores moleculares fueron calculados utilizando RDkit. Los descriptores usados se seleccionaron en función de su variabilidad y el coeficiente de Pearson para desarrollar modelos de predicción de SVM, RF, DT, XGB, k-NN y NB en el entrenamiento y validación mediante datos test.*

Naive Bayes

Los métodos bayesianos se basan en el teorema de Bayes, describiendo la probabilidad de un evento que podría haber sido el resultado de cualquiera de dos o más causas.

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

Esta ecuación expresa la probabilidad de un evento A, dando un evento B ya ocurrido. Esto se conoce como probabilidad condicional, donde la probabilidad de A depende de lo que haya pasado en el evento B. Por tanto, los métodos bayesianos se pueden emplear para modelar las dependencias entre variables que se influyen directamente entre sí (Dempster, 1968).

Por tanto, se trata de un modelo simple, rápido y eficaz, que trabaja bien con datos ruidosos y perdidos, necesita pocos datos de entrenamiento y es fácil de obtener la probabilidad estimada para una predicción. En contra, a veces recae en una suposición errónea de igualdad, no es buena para set de datos con muchas variables numéricas y las probabilidades estimadas son menos fiables que las predichas (Lantz, 2015).

Para este modelo se empleó la función naiveBayes() del paquete e1071.

El parámetro Laplace se encarga de agregar un pequeño número a cada uno de los conteos de frecuencia, asegurando que cada característica tiene una probabilidad distinta de 0 de ocurrir con cada clase (Meyes et al., 2022).

Support Vector Machine

Las máquinas de vectores de soporte tratan de crear un límite plano llamado hiperplano, el cual divide el espacio para crear una partición que divide los datos en grupos de valores de clase similares. El hiper plano de margen máximo (MMH) crea la mayor separación entre clases, mejorando la posibilidad de que, a pesar de los descriptores aleatorios, los datos permanecerán en el lado correcto del límite. Los vectores de soporte son los valores de cada clase que se encuentran más próximos al MMH, donde al menos debe de haber un vector de soporte por cada tipo de clase (Vapnik, 2000).

En caso de que la información no se pueda dividir de forma lineal, se implementa el uso de kernels que añaden nuevas dimensiones a la información para poder crear nuevas separaciones, permitiendo al modelo aprender conceptos que no estaban medidos en los datos originales (Lantz, 2015).

Los SVM se adaptan a cualquier tipo de tarea de aprendizaje, tanto de clasificación como de predicción numérica, no se ven afectados por el ruido ni por sobreajuste, pero es complicado encontrar la mejor combinación y el entrenamiento puede ser lento (Lantz, 2015).

En este estudio se emplean dos tipos de kernel:

- Kernel lineal: donde no se produce transformación de los datos.

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i} \cdot \overrightarrow{x_j})$$

- Kernel Gaussiano RBF: coloca una función de base radial centrada en cada punto, realizando modificaciones lineales para asignar puntos a espacios de mayor dimensión que son más fáciles de separar.

$$K(\overrightarrow{x_i}, \overrightarrow{x_j}) = e^{-||\overrightarrow{x_i} - \overrightarrow{x_j}||^2 / 2\sigma^2}$$

Para la función ksvm kernel lineal es igual a “vanilladot” y kernel radial es igual a “rbfdot”.

Para el valor C, que es el coste de violación de las restricciones, se le dio valores de 1, 3 y 7.

Decision Tree

El algoritmo C5.0 se ha convertido en el estándar para la producción de árboles de decisión, logrando resultados similares a SVM, pero siendo más fáciles de entender y desarrollar. Funciona bien para la mayoría de los problemas de clasificación, con alto porcentaje de aprendizaje automático y con capacidad de trabajar con datos nominales, resultando en un modelo con interpretación sin necesidad de conocimiento matemático. Sus problemas son principalmente la facilidad al sobreajuste y que pequeños cambios en el set de datos pueden resultar en grandes cambios en las decisiones logísticas (Lantz, 2015).

Empezando desde la raíz el árbol se divide en dos o más ramas las cuales a su vez también se pueden dividir, hasta que se llega a una hoja, que es un nodo que no se divide más. A cada hoja se le asigna una propiedad, mientras que los nodos fuera de la hoja se convierten en una condición de prueba. Por tanto, un compuesto se clasifica en función del nodo de la hoja, después de atravesar una serie de preguntas (nodos) y respuestas (decide qué camino toma) (Lavecchia, 2015).

Para este trabajo, se empleó la función c5.0() con valor para el parámetro coste su valor por defecto NULL, por lo que no existen costes asociados y un valor de trials, es decir, de número de iteraciones para la mejora de los resultados de 1, 15, 40 y 50.

Random Forest

El algoritmo RF es un método supervisado, donde se crea un “bosque” desde varias perspectivas para hacerlo aleatorio. Para las tareas de clasificación, la salida es la clase seleccionada para la mayoría de los árboles. Para las tareas de regresión se devuelve la predicción media de los árboles individuales (Dara et al., 2022).

RF corrige el hábito de los árboles de decisión de sobreajustarse, puede trabajar con ruido y datos perdidos, seleccionando los descriptores más importantes, siendo difícil de interpretar y con dificultades a la hora de ajustar el modelo a los datos (Lantz, 2015).

Para la función randomForest(), se le dio al parámetro mtry (número de variables para muestrear aleatoriamente como candidatas en cada división) el valor por defecto que en este caso es igual a 7 y, para ntree que es el número de veces que se predicen los datos, unos valores de 50, 100, 150 y 200.

Extreme Gradient Boosting

XGB es un algoritmo basado en árboles, útil para problemas de clasificación y regresión. Utiliza su propio método de creación de árboles donde los resultados de similitud y ganancia determinas las divisiones de los nodos (Sheridan et al., 2016).

$$Similarity\ Score = \frac{(\sum_{i=1}^{n} Residual_i)^2}{\sum_{i=1}^{n} [Previous\ Probability_i * (1 - Previous\ Probability_i)] + \lambda}$$

Los residuos son los valores observados, lambda regulariza la influencia de las pequeñas hojas. La probabilidad previa es la probabilidad de un evento calculada en un paso anterior. Para cualquier árbol la probabilidad anterior se vuelve a calcular en función de la predicción inicial y las predicciones anteriores (towardsdatascience.com).

Para la función xgboost(), se tuvo en cuenta el parámetro max.depth, que es la profundidad máxima de un árbol (a mayor nivel, mayor complejidad del modelo) con valores de 6 y 12; y nrounds que es el número de iteraciones que se realizan para mejorar el modelo al que se le dio valores 1, 6 y 12 (xgboost.readthedocs.io).

3.3. Evaluación de los Modelos

La calidad de los modelos de entrenamiento fue medida por la cantidad de falsos positivos (FP), falsos negativos (FN), verdaderos positivos (VP), verdaderos negativos (VN), tasa de error (TE), valor de kappa (ajusta la precisión teniendo en cuenta la posibilidad de una predicción correcta solo por casualidad), sensibilidad (número de ejemplos positivos que fueron clasificados correctamente, SE), especificidad (número de ejemplos negativos clasificados correctamente, SP), precisión (P) y Área debajo de la curva (AUC), (Lantz, 2015), las cuales están dadas por las ecs 1-4.

(1) $SE = VP/VP + FN$

(2) $SP = VN/VN + FP$

(3) $P = (VP + VN)/(VP + VN + FN + FP)$

(4)

$$\frac{\sum_{i=1}^{q}\sum_{j=1}^{q} I[y_i < y_j]I'[a_i < a_j]}{\sum_{i=1}^{q}\sum_{j=1}^{q} I[y_i < y_j]},$$

$$I'[a_i < a_j] = \begin{cases} 0, & a_i > a_j, \\ 0.5 & a_i = a_j, \\ 1, & a_i < a_j, \end{cases} \qquad I[y_i < y_j] = \begin{cases} 0, & y_i \geq y_j, \\ 1, & y_i < y_j, \end{cases}$$

a_i – the response of the algorithm on the a-th object y_i – its label (class) q – the number of objects in the test

VP representa el número de moléculas activas inhibitorias que son predichas como inhibitorias. VN representa ele número de compuestos inactivos que son predichos como inactivos. FP es el número de componentes inactivos que han sido predichos como activos. FN es el número de moléculas inhibitorias que han sido predichas como inactivas.

Los modelos fueron puestos aprueba a partir de los datos del set de datos test.

3.4. Mejora de hiperparámetros del modelo seleccionado.

Se utilizó la metodología de 10-fold cross-validation, un procedimiento utilizado para estimar la habilidad del modelo en nuevos datos. Se trata de un remuestreo con un solo parámetro llamado k que se refiere a la cantidad de grupos en los que se dividirá una muestra dada. Dado el modelo se realizó una búsqueda de la mejor combinación de hiperparámetros para lograr el mayor valor de precisión del modelo (Anguita et al., 2009).

4. Resultados y Discusión

La comparación entre los diferentes modelos para la selección del mejor candidato se realizó teniendo en cuenta el número de falsos positivos, de falsos negativos, verdaderos positivos, verdaderos negativos, tasa de error, valor de kappa, sensibilidad, y precisión. Siempre se tiene como positivo aquella molécula que presenta actividad frente a CHIKV, es decir, “Activity” = 1.

4.1. k-Nearest Neighbor

Para el modelo k-NN (Venables & Ripley, 2002), se dieron para *k* (número de vecinos a considerar) valores de 1, 5 y 11; obteniendo los siguientes resultados:

Tabla 3. *Resultados obtenidos del modelo kNN para valores de k de 1,5 y 11. K (número de vecinos a considerar), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

K	VN	VP	FN	FP	TE	P	Kappa	SE	SP
1	31	29	14	16	0.3333	0.6667	0.3333	0.6744	0.6596
5	29	27	16	18	0.3778	0.6222	0.2444	0.6279	0.6170
11	29	26	17	18	0.3889	0.6111	0.2215	0.6047	0.6170

Cómo se observa en la Tabla 3, los mejores resultados de predicción se obtuvieron con un valor de k=1, dónde el modelo predijo 31 moléculas sin actividad y 29 moléculas con actividad correctamente, siendo, por tanto, los valores más altos de especificidad (0.6596) y sensibilidad (0.6744) respectivamente. Con esta capacidad se logró una capacidad de precisión de 0.6667. Los valores son relativamente parecidos a excepción del valor kappa (0.3333), dónde saca casi 0.1 al siguiente mejor valor, hablando así de un arreglo justo (Tabla 3) (Lantz, 2015).

4.2. Naive Bayes

En el modelo NB, se dio para el estimador Laplace (añade un pequeño número a cada valor en la tabla de frecuencias para que la probabilidad en ninguna de las clases sea 0) (Lantz, 2015; Meyes et al., 2022), valores de 0, 1 y 5; obteniendo los siguientes resultados:

Tabla 4. *Resultados obtenidos del modelo NB para valores de laplace de 0,1 y 5. Laplace (valor añadido a las tablas de frecuencia), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

Laplace	VN	VP	FN	FP	TE	P	Kappa	SE	SP
0	36	22	21	11	0.3556	0.6444	0.2804	0.5116	0.7660
1	36	22	21	11	0.3556	0.6444	0.2804	0.5116	0.7660
5	36	22	21	11	0.3556	0.6444	0.2804	0.5116	0.7660

En todos los casos se obtuvieron los mismos resultados con un alto porcentaje de moléculas sin actividad predichas correctamente (especificidad de 0.7660) y un número más bajo de moléculas con actividad predichas (22) lo que produce un valor de sensibilidad total de 0.5116. El número de falsos negativos es muy elevado (21), mientras que el de falsos positivos es bastante inferior (11), por lo que se observa una cierta inclinación por parte del modelo a la hora de predecir, a clasificar las moléculas como no activas. Esto ha resultado en una tasa de error de 0.3556, una precisión de 0.6444 y un valor kappa justo de 0.2804 (Tabla 4).

En este caso viendo que el estimador Laplace no ha producido ningún cambio en la predicción, se tomará Laplace de valor 0 como referencia en la comparativa con el resto de los modelos.

4.3. Support Vector Machine

A continuación, en el modelo SVM se tuvo en cuenta el kernel (vanilladot: lineal y rbfdot: núcleo de base radial “Gaussiano”) y también el parámetro coste (C) que es valor de coste de violación de las restricciones, que se le dieron valores de 1,3 y 7 para ambos casos de kernel (Chan C-C & Lin C-J, 2001).

Tabla 5. *Resultados obtenidos del modelo SVM para kernel (transformación y combinación de vectores) lineal o Gaussiano, valores de C de 1,3 y 7. C (coste de violación de las restricciones), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

Kernel	C	VN	VP	FN	FP	TE	P	Kappa	SE	SP
Lineal	1	31	30	13	16	0.3222	0.6778	0.3562	0.6977	0.6596
	3	30	26	17	17	0.3778	0.6222	0.2429	0.6047	0.6383
	7	29	28	15	18	0.3667	0.6333	0.2674	0.6512	0.6170
Gaussiano	1	29	27	16	18	0.3778	0.6222	0.2444	0.6279	0.6170
	3	32	25	18	15	0.3667	0.6333	0.263	0.5814	0.6809
	7	32	28	15	15	0.3333	0.6667	0.332	0.6512	0.6809

A razón de los datos observados en la Tabla 5, se deduce que los mejores parámetros para el modelo SVM fueron un kernel lineal y un coste C de 1. Con estos parámetros se lograron un total de 31 verdaderos negativos y 13 falsos negativos, lo que resulta en una especificidad del 0.6596. En el caso de las moléculas que presentan actividad, 30 de ellas fueron clasificadas correctamente, mientras que se produjeron 16 falsos positivos, dando una sensibilidad de 0.6977. Esta configuración obtuvo una precisión del 0.6778 y por tanto una tasa de error de 0.3222 con un ajuste kappa de 0.3562, próximo a un arreglo moderado.

Una configuración con resultados similares fue el modelo gaussiano con coste 7, que obtuvo una mayor especificidad, pero con una precisión, una sensibilidad y un valor kappa inferior a la configuración seleccionada.

4.4. Decision Tree

En el caso de los DT se tuvo en cuenta el método de boosting que se basa en la combinación de numerosos intentos de aprendizaje “débiles”, para crear uno por sí solo, mucho mejor que cualquiera de los otros por separado (Lantz, 2015). En esta función, ese parámetro recibe el nombre de trials, que es el número de iteraciones de refuerzo para mejorar los datos del modelo (Quinlan R, 1993). Para este caso, se le dio al parámetro trials los valores 1, 15, 40 y 50 obteniendo los resultados que se ven a continuación (Tabla 6):

Tabla 6. *Resultados obtenidos del modelo DT para el parámetro trial valores de 1, 15, 40 y 50 respectivamente. Trials (número de iteraciones de refuerzo), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

Trials	VN	VP	FN	FP	TE	P	Kappa	SE	SP
1	29	29	14	18	0.3556	0.6444	0.2903	0.6744	0.6710
15	33	30	13	14	0.3	0. 7	0.3994	0.6977	0.7021
40	33	31	12	14	0.2889	0.7111	0.4222	0.7209	0.7021
50	31	30	13	16	0.3222	0.6778	0.3562	0.6977	0.6596

Se observa que en el caso para un valor de trials de 40, el modelo consiguió una precisión del 0.7111, logrando un total de 33 verdaderos negativos y 31 falsos negativos, con una especificidad de 0.7021 y una sensibilidad de 0.7209. Esto hace que el número de falsos positivos y falsos negativos se viese reducido a 14 y 12 respectivamente, logrando una tasa de error de 0.2889. Además, se logró un valor de kappa de 0.4222, por lo que se trata de un arreglo moderado.

Con 15 iteraciones se logró la misma especificidad, pero con un falso positivo más. Al seguir aumentando el número de iteraciones, el modelo pierde capacidad de predicción obteniendo peores resultados, pero mejorando el primer intento sin la opción de boosting.

4.5. Random Forest

Por otro lado, se empleó el parámetro ntree (número de veces que se predicen los datos de entrada) de RF, al que se le dio valores de 50, 100, 150 y 200 árboles (Breiman, 2001).

Tabla 7. *Resultados obtenidos del modelo RF para el parámetro ntree valores de 50, 100, 150 y 200. Ntree (número de árboles), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

ntree	VN	VP	FN	FP	TE	P	Kappa	SE	SP
50	34	31	13	12	0.2778	0.7222	0.4439	0.7045	0.7391
100	33	32	14	11	0.2778	0.7222	0.445	0.6957	0.7500
150	33	31	14	12	0.2889	0.7111	0.4222	0.6889	0.7333
200	33	29	14	14	0.3111	0.6889	0.3765	0.6744	0.7021

Como se puede observar en la Tabla 7, tanto para un número de 50 árboles como para 100, se ha obtenido una precisión de 0.7222. En un caso se ha obtenido 34 verdaderos negativos por los 33 con ntree igual 100. Del mismo

modo con valor 100 se obtiene un verdadero positivo más que con 50, hablando de un total de 32. El valor de ntree 50 presenta 13 falsos negativos y 12 falsos positivos con una sensibilidad y especificidad de 0.7045 y 0.7391, respectivamente. En el caso de usar 100 árboles 14 han sido los falsos negativos registrado por 11 falsos positivos, con una sensibilidad de 0.6957 y 0.75 de especificidad. Ambas opciones son válidas, pero en este caso se escogerá el modelo con el parámetro de árboles de valor 100, puesto que el valor kappa, siendo un ajuste moderado en ambos, es ligeramente superior en este caso (0.445 frente a 0.4439).

En cualquier caso, con los diferentes parámetros empleados, se han logrado unos ajustes moderados con buenos datos de precisión.

4.6. Extreme Gradient Boosting

Por último, en el modelo XGB para la predicción de actividad de las moléculas de estudio, se tuvo en cuenta los parámetros max.depth (es la profundidad máxima de un árbol, a mayor valor más complejo se vuelve el modelo, pero sufre riesgo de sobreajuste) y nrounds (número de iteraciones que se realizan para mejorar el modelo) (xgboost.readthedocs.io). A max.depth se le dieron valores de 6 y 12, mientras que nrounds tuvo valores de 1, 6 y 12.

Tabla 8. *Resultados obtenidos del modelo XGB para los parámetros max.depth valores de 6 y 12; y nrounds 1, 6 y 12. Max.depth (profundidad máxima), nrounds (número de iteraciones de refuerzo), VN (verdaderos negativos), VP (verdaderos positivos), FN (falsos negativos), FP (falsos positivos), TE (tasa de error), P (precisión), Kappa, SE (sensibilidad) y SP (especificidad).*

Max.depth	nrounds	VN	VP	FN	FP	TE	P	Kappa	SE	SP
6	1	33	24	19	14	0.3667	0.6333	0.2616	0.5581	0.7021
	6	31	32	11	16	0.3	0.7	0.4018	0.7442	0.6596
	12	32	30	13	15	0.311	0.6889	0.3778	0.6977	0.6809
12	1	33	24	19	14	0.3667	0.6333	0.2616	0.5581	0.7021
	6	33	24	19	14	0.3667	0.6333	0.2616	0.5581	0.7021
	12	29	33	10	18	0.311	0.6889	0.3778	0.7674	0.6170

Los mejores rendimientos del modelo se obtuvieron con unos reglajes de mx.depth y nrounds igual a 6 (Tabla 8). La tasa de error fue del 0.3 lo que supone una precisión del 70%. Se consiguió una sensibilidad muy alta de 0.7442 debido al alto número de verdaderos positivos (32) y al bajo número de falsos negativos (11). Por otro lado, el nivel de especificidad fue uno de los más bajos con respecto al resto de configuraciones (0.6596), lo que puede indicar que, con estos valores en los parámetros, el modelo sufra una ligera tendencia a predecir las moléculas como positivas frente a CHIKV (como pasa con la configuración

max.depth 12 y round 12). El valor de kappa es de 0.4018 por lo que se trata de un ajuste moderado y es el único reglade de parámetros que tiene ese nivel de ajuste respecto del resto.

4.7. Selección del Modelo

Una vez se tuvo en cuenta los diferentes parámetros y sus resultados para cada modelo y la selección de la mejor configuración para cada uno de estos, se realizó una comparativa de los diversos modelos para su uso como método de predicción de moléculas con actividad de inhibición frente al CHIKV.

Tabla 9. *Resultados de la mejor configuración para cada modelo estudiado.*

Modelo	VN	VP	FN	FP	TE	P	Kappa	SE	SP
k-NN	31	29	14	16	0.3333	0.6667	0.3333	0.6744	0.6596
NB	36	22	21	11	0.3556	0.6444	0.2804	0.5116	0.766
SVM	31	30	13	16	0.3222	0.6778	0.3562	0.6977	0.6596
DT	33	31	12	14	0.2889	0.7111	0.4222	0.7209	0.7021
RF	33	32	14	11	0.2778	0.722	0.445	0.6957	0.75
XGB	31	32	11	16	0.3	0.7	0.4018	0.7442	0.6596

El modelo que presentó mayor sensibilidad fue XGB con un valor de 0.7442 (Figura 11) debido a que, junto a RF, fue la que mayor número de verdaderos positivos tuvo con un total de 32 y con 11 obtuvo el menor número de falsos negativos (Tabla 9). Tras este modelo, el DT logró una sensibilidad de 0.7209, seguido muy de cerca por SVM y RF con 0.6977 y 0.6957 (Figura 11) respectivamente. El modelo NB fue el que peor resultados de sensibilidad mostró con un valor de 0.5116 (Figura 11), por esa tendencia mencionada de manera previa, a clasificar las moléculas como no activas frente al virus.

Debido a esto, NB presentó los mejores datos de especificidad con un valor de 0.766 (Figura 12), gracias a los 36 verdaderos negativos y a sus 11 falsos positivos, los mismos logrados por el modelo RF (Tabla 9), que lo colocó como el segundo con mejores datos de especificidad (0.75) (Figura 12). A continuación, los DT obtuvieron un valor de 0.7021 (Figura 12) y, por último, tanto el modelo de k-NN, como el SVM y XGB dieron de resultado 0.6596 con el mismo número de verdaderos negativos y falsos positivos (31 y 16, respectivamente) (Tabla 9).

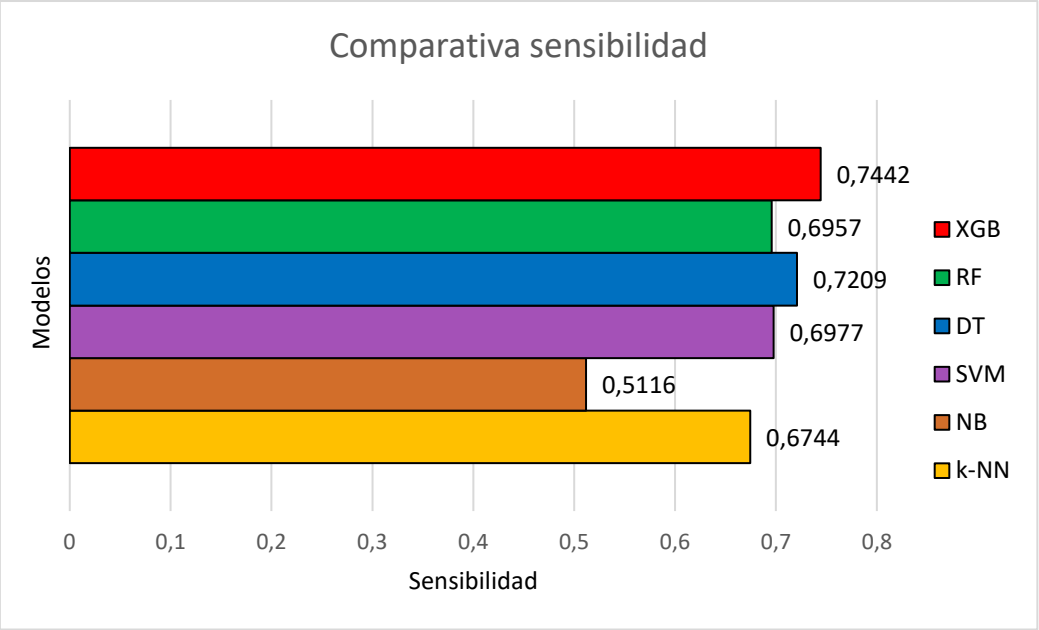

Figura 11. *Gráfica comparativa de los resultados de sensibilidad obtenidos en cada modelo.*

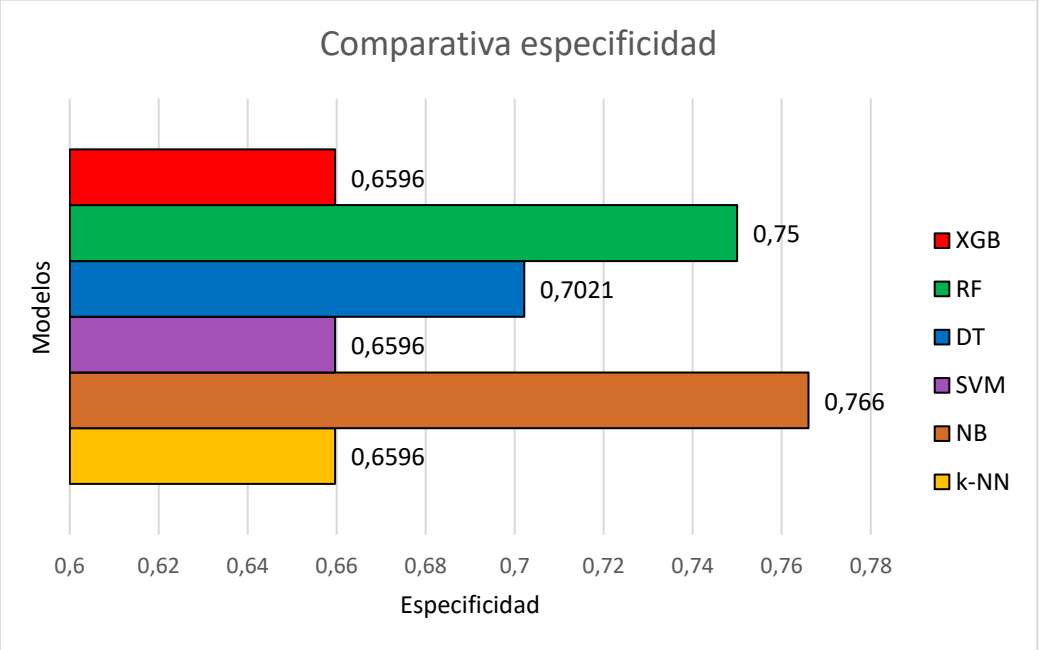

Figura 12. *Gráfica comparativa de los resultados de especificidad obtenidos en cada modelo.*

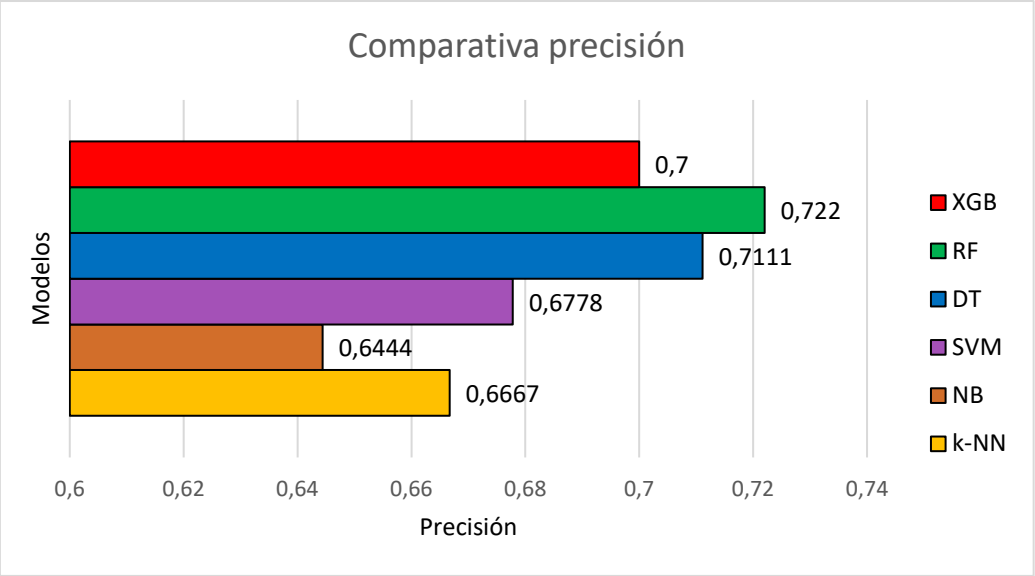

Figura 13. *Gráfica comparativa de los resultados de precisión obtenidos en cada modelo.*

Estos resultados hicieron, que RF fuera el modelo de mayor precisión con un valor de 0.7222. Resultados muy similares obtuvieron los modelos de DT y XGB con 0.711 y 0.7 respectivamente. El modelo de NB consiguió los peores datos de precisión con un valor de 0.6444, lo que supone algo más de un fallo en 1/ 3 moléculas predichas (Figura 13).

Relacionados con la precisión, los valores de kappa mostraron el mismo patrón de distribución. El modelo RF fue el que mejor valor obtuvo de un total de 0.445 tratándose de un arreglo moderado. Tanto XGB como DT, lograron también un arreglo moderado de 0.4018 y 0.4222 respectivamente. Por otro lado, tanto k-NN (0.3333) y SVM (0.3562) sufrieron un arreglo justo y más por debajo, NB logró un valor kappa de 0.2804 (Figura 14).

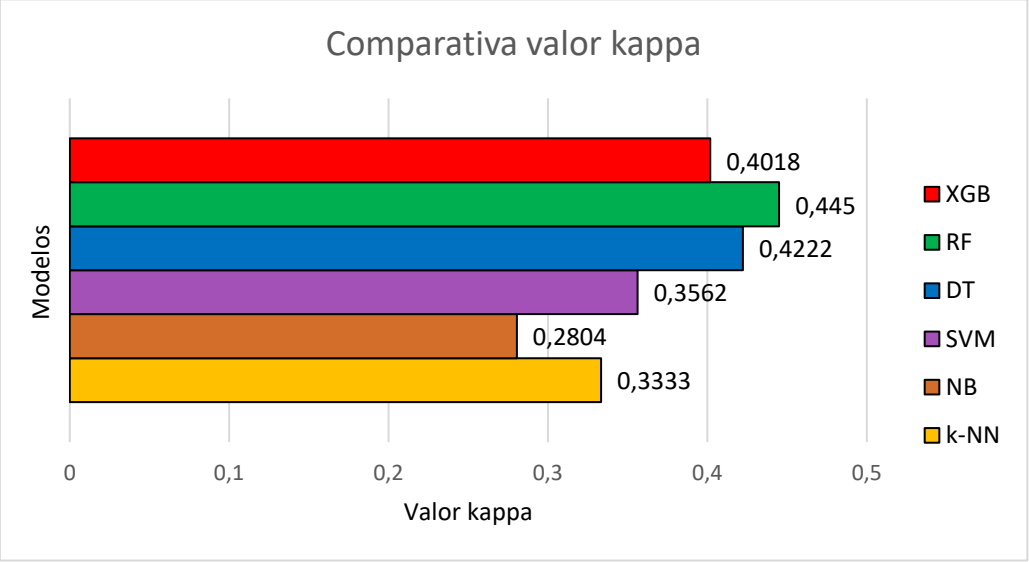

Figura 14. *Gráfica comparativa de los resultados de kappa obtenidos en cada modelo.*

Por último, se realizó el cálculo de la curva ROC y el AUC de cada modelo para comprobar si realmente tenían valor predictivo. Los datos obtenidos fueron que RF es el modelo que mayor valor predictivo presenta con un buen valor de 0.786, seguido por los modelos DT y XGB con valores justos de 0.765 y 0.761 respectivamente. El modelo NB presenta un valor de AUC aceptable de 0.74, mientras que tanto k-NN (0.667) como SVM (0.690) presentaron un poder de predicción pobre (Figura 15).

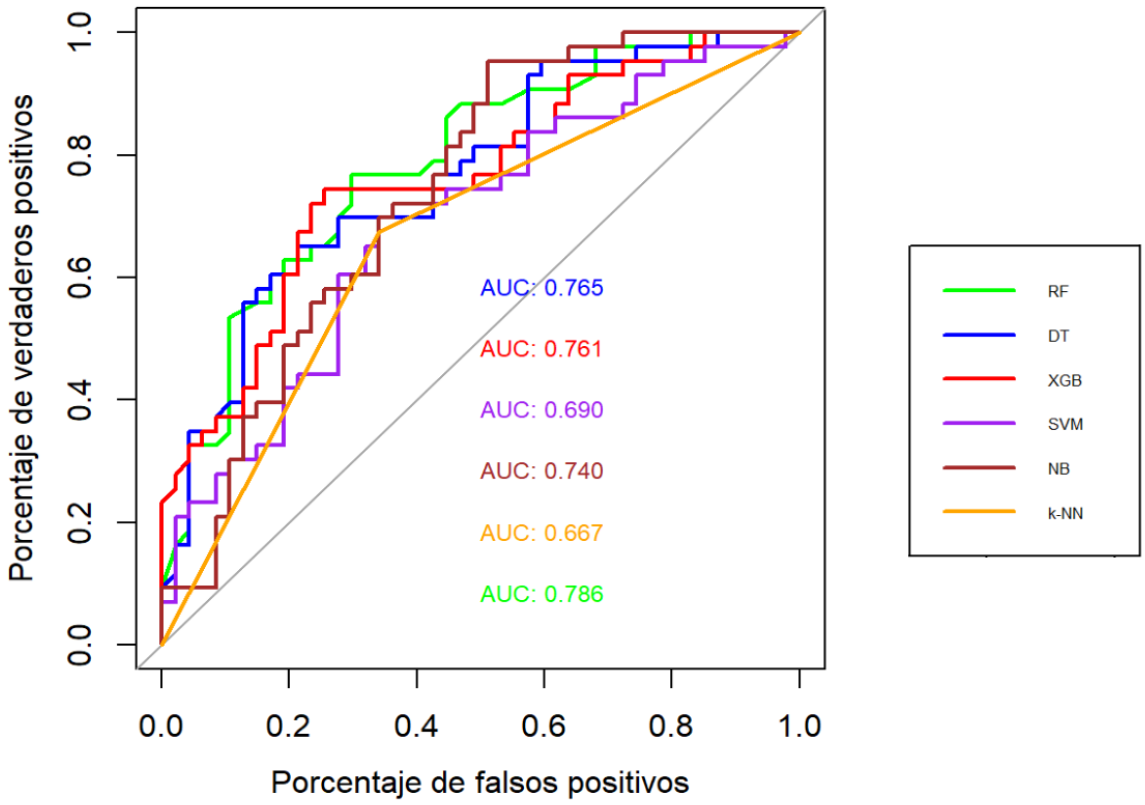

Figura 15. *Gráfica comparativa de la curva ROC y el valor AUC de los diferentes modelos de estudio.*

Por los motivos expuestos a lo largo de estos resultados, se seleccionó al modelo RF como el mejor modelo para la predicción de actividad frente al chikungunya de nuevas moléculas.

RF es capaz de clasificar las variables según el peso que tienen estas a la hora de tomar las decisiones en la predicción. Para este modelo los descriptores SlogP, smr_VSA3, Energy, peoe_VSA7 y peoe_VSA8 (Figura 16) fueron los descriptores de mayor importancia.

Mediante estos descriptores se pudo comprobar si existe cierta tendencia por parte de las moléculas con actividad o sin actividad a dar ciertos valores para cada descriptor. Se tomaron los 8 descriptores más importantes obtenidos en la Figura 16 y se realizó un gráfico boxplot.

Los valores de los descriptores de la Figura 17 se mostraron distribuidos de manera similar para ambos tipos de moléculas. Sin embargo, en la Figura 18 se pueden observar ligeras diferencias entre los valores de la variable a predecir Activity y que se vieron claramente reflejadas en el gráfico smr_VSA3 donde parece existir una cierta tendencia a que las moléculas con actividad frente a CHIKV presenten valores más bajos de este descriptor.

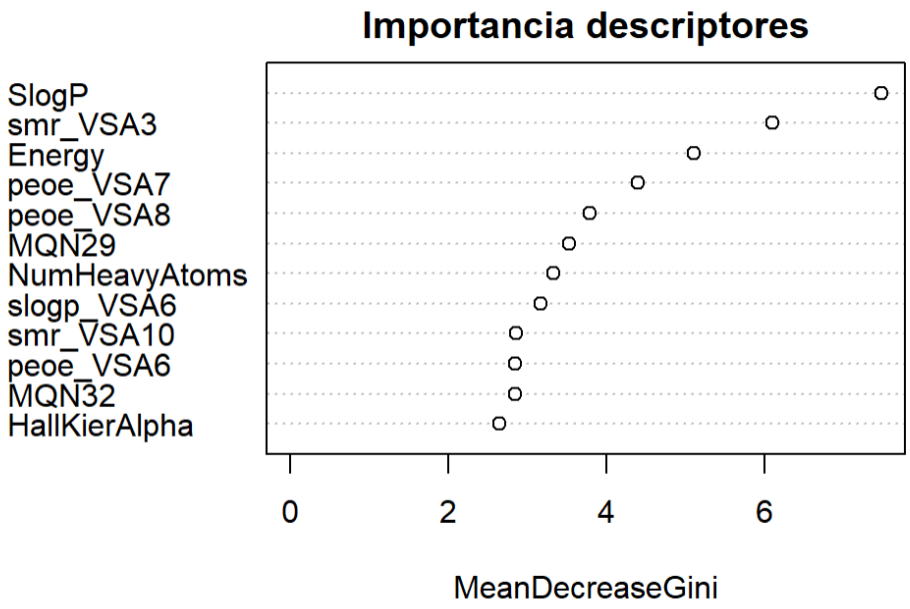

Figura 16. *Gráfica comparativa de la importancia de los descriptores para el modelo RF, usando el Gini de descenso medio que mide qué tan puros son los nodos al final de árbol sin cada variable.*

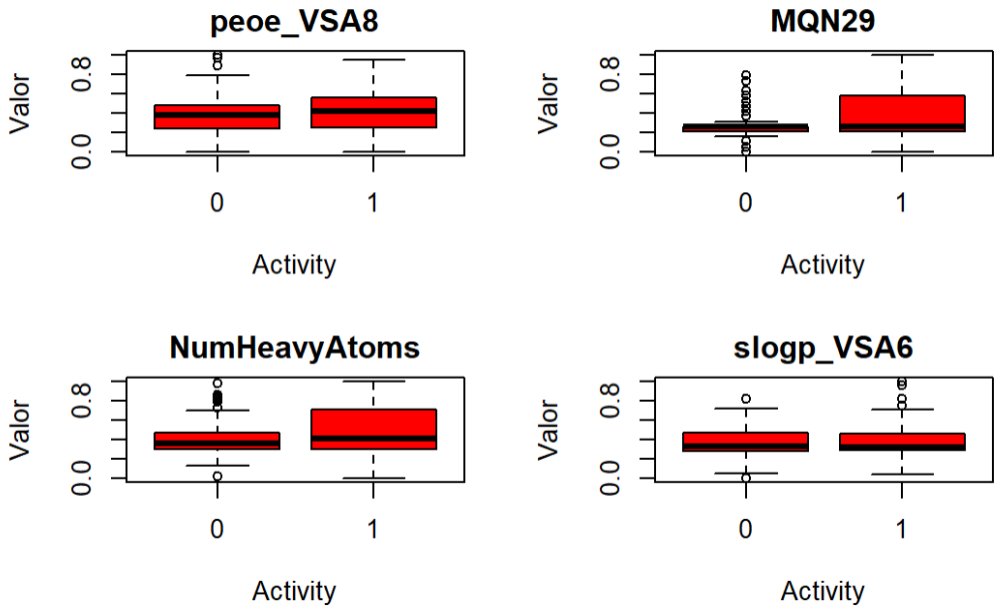

Figura 17. *Boxplot de la distribución de valores de los descriptores peoe_VSA8, MQN29, NumHeavyAtoms, slogp_VSA6 para el modelo RF teniendo en cuenta la actividad o no actividad de las moléculas de estudio.*

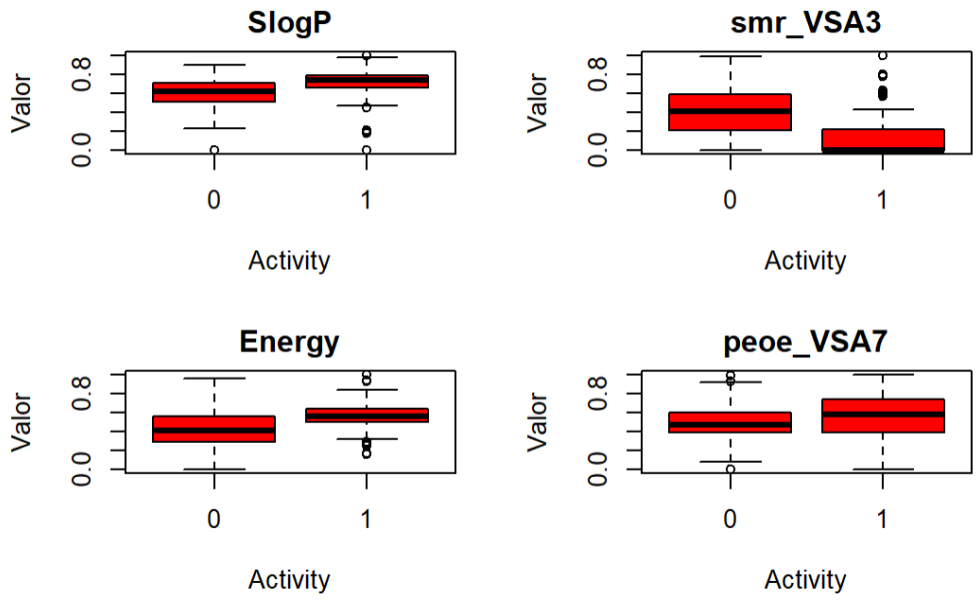

Figura 17. *Boxplot de la distribución de valores de los descriptores SlogP, smr_VSA3, Energy y peoe_VSA7 para el modelo RF teniendo en cuenta la actividad o no actividad de las moléculas de estudio.*

La selección no significa que los modelos de RF sean lo modelos de referencia para la búsqueda de nuevos fármacos, sino que, en la mayoría de los casos, la actuación de estos modelos dependerá de los datos empleados y su variable a predecir.

Gawriljuk y su equipo trabajaron en el desarrollo de un nuevo componente antiviral frente al virus de la fiebre amarilla, obteniendo mejores resultados en modelos de k-NN con una precisión del 0.85 y un valor AUC de 0.77, o de SVM con 0.82 y 0.81 respectivamente, que los obtenidos con RF con una precisión de 0.83 y un AUC de 0.77 (Gawriljuk et al., 2021).

Kamboj et al, emplearon en 2022 modelos de SVM, RF, k-NN y ANN, para descubrir moléculas capaces de actuar sobre proteínas no estructurales del virus de la hepatitis C. Los resultados que obtuvieron fueron que para un algoritmo de selección de Support Vector Regression, los valores del coeficiente de regresión (R^2) fueron de 0.72 para SVM, 0.62 para k-NN y RF; y 0.71 para ANN (Kamboj et al., 2022).

En un ensayo para la predicción de inhibidores de BuChe (objetivo farmacológico para el tratamiento del Alzheimer) se emplearon modelos de SVM y NB logrando sorprendentes niveles de predicción de 98.12 y 86.41 para una serie de variables obtenidas de descriptores MOE (Fang et al., 2013).

Para la predicción de inhibición de proteínas resistentes al cáncer de mama, que facilitarían la evaluación a posibles resistencias frente a otros medicamentos en los primeros pasos del diseño de fármacos, Jiang y su grupo emplearon múltiples modelos en los que se encuentran SVM, XGB, k-NN o NB. Los mejores resultados de precisión los obtuvo el modelo SVM con una precisión de 0.911 y

un AUC de 0.958 para el set de datos de testeo. Ligeramente por debajo se encontraba XGB con una precisión de 0.891 y un AUC de 0.957. Los valores de precisión descendían notablemente para k-NN (0.857) y NB (0.78) en comparación a los dos primeros modelos (Jiang et al., 2020).

Por último, un estudio de comparación de los modelos RF y SVM para la predicción de protección frente a la radiación y toxicidad de diferentes moléculas con propiedades farmacológicas. Por un lado, lo que se obtiene es que el modelo RF para las propiedades de toxicidad es mejor que el modelo SVM obteniendo unos valores AUC de 0.778 frente a los 0.716 del otro modelo. Por otro lado, en el caso de las propiedades de protección frente a la radiación es el modelo SVM el que obtiene mejor valor AUC (0.646 frente al valor 0.619 de RF) (Matsumoto et al., 2016).

Por tanto, a la hora de estudiar moléculas como posibles inhibidores de una enfermedad, no se puede tomar un único modelo como referencia, sino que será necesario realizar varias pruebas con diferentes modelos (Fang et al., 2013; Gawriljuk et al., 2021; Jiang et al., 2020; Kamboj et al., 2022; Matsumoto et al., 2016), puesto que, tanto los descriptores que se han empleado como la variable o variables a predecir, afectan de diferente manera en la capacidad de predicción de los modelos.

4.8. Optimización de los hiperparámetros

Tras seleccionar el modelo RF, como el mejor método de predicción de actividad inhibitoria frente al virus del chikungunya, se quiso comprobar el set de hiperparámetros para obtener la mayor precisión posible.

Para ello se tuvieron en cuenta los parámetros mtry (número de variables que se muestrean aleatoriamente como candidatos en cada división) y ntree que como se ha explicado con anterioridad, es el número de veces que se predicen los datos de entrada.

Mediante 10-fold cross validation se obtuvo, que el mejor set de hiperparámetros era con mtry un valor de 22 y ntree de valor 75 alcanzando una precisión de 0.7011696 en comparación a los 0.6847953 obtenidos de la configuración de hiperparámetros ntree = 100 y mtry=7 (hiperparámetros del modelo seleccionado).

5. Conclusiones y trabajos futuros

En este estudio, se ha desarrollado una serie de algoritmos de regresión mediante técnicas de machine learning: SVM, RF, k-NN, DT, NB y XGB, para crear un sistema de predicción de nuevas moléculas anti-CHIKV capaces de inhibir su actividad, a partir del entrenamiento de una serie de descriptores obtenidos mediante RDkit.

Los modelos de predicción actuaron de forma correcta con valores de precisión entre 0.6444 a 0.722 y valores AUC de 0.667 a 0.786 para los datos test. Tras el análisis exhaustivo de la sensibilidad, especificidad, valor kappa, precisión y AUC de los modelos se tomó al modelo RF con valor ntree = 75 como el modelo referencia en este estudio.

Al modelo de referencia se le realizó una optimización de sus hiperparámetros, pasando de una precisión de 0.68479 a una precisión de 0.70117.

Con todos los resultados obtenidos, se puede considerar que se han cumplido con los objetivos previstos al inicio del proyecto, puesto que a partir de una serie de moléculas de una base de datos de ChEBML, hemos sido capaces de construir mediante sus descriptores, un método de predicción de moléculas con actividad inhibitoria frente al CHIKV, que actualmente no tiene ningún tipo de vacuna o medicamento dirigido específicamente y, por tanto, supone un riesgo para las sanidades públicas, tanto en países desarrollados como en vías de desarrollo, por la posibilidad de saturación de hospitales, como se ha vivido recientemente con la COVID 19.

En caso de plantear este método para la búsqueda de moléculas inhibitorias, veo necesario una optimización de los métodos, centrada en el uso de nuevos descriptores. Por la limitación del tiempo no ha sido posible, pero se podrían haber comparado diferentes softwares de obtención de descriptores como realizaron Fang J et al, 2013, empleando a parte de RDkit otros como ADRIANA.Code, MOE software o Discovery Studio, incrementando el número de descriptores que se obtienen para cada molécula, lo que podría suponer una mejora en los resultados de predicción de los modelos.

Para finalizar, este TFM es un primer informe utilizando aproximaciones de machine learning. Se podría continuar realizando predicciones sobre medicamentos en uso para comprobar si estos pudieran presentar actividad anti-CHIKV y comprobar el funcionamiento del modelo mediante medicamentos que ya han sido probados y presentan o no actividad. Otro paso podría ser la validación de los medicamentos predichos mediante molecular docking, pero actualmente sigue faltando información acerca de las vías de como estos posibles medicamentos afectan en el desarrollo del virus.

6. Glosario

AUC: Area under the ROC curve

CHIKV: Chikungunya virus

DENV: Dengue virus

DT: Decision tree

EC50: Half maximal effective concentration

k-NN: K-nearest neighbor

MOE: 3D molecular descriptors, which are dependent on the conformation of a molecule, include potential energy descriptors, surface area, volume, shape descriptors, and charge descriptors.

NB: Naive Bayes

NSAIDs: Non-steroidal anti-inflammatory drugs

ORF: Open reading frame

RF: Random Forest

ROC: Receiver Operator Characteristic

SVM: Support vector machine

XGB: Extreme gradient boosting

ZIKV: Zika virus

7. Bibliografía

Abdelnabi, R., Amrun, S. N., Ng, L. F. P., Leyssen, P., Neyts, J., & Delang, L. (2017). Protein kinases C as potential host targets for the inhibition of chikungunya virus replication. *Antiviral Research*, *139*, 79–87. https://doi.org/10.1016/j.antiviral.2016.12.020

Albulescu, I. C., White-Scholten, L., Tas, A., Hoornweg, T. E., Ferla, S., Kovacikova, K., Smit, J. M., Brancale, A., Snijder, E. J., & van Hemert, M. J. (2020). Suramin inhibits chikungunya virus replication by interacting with virions and blocking the early steps of infection. *Viruses*, *12*(3). https://doi.org/10.3390/v12030314

Anguita, D., Ghio, A., Ridella, S., & Sterpi, D. (2009). K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines. *Proceedings of The 2009 International Conference on Data Mining, DMIN, Las Vegas, USA.*

Bakar, F. A., & Ng, L. F. P. (2018). Nonstructural proteins of alphavirus—potential targets for drug development. *In Viruses, 10*(2). https://doi.org/10.3390/v10020071

Battisti, V., Urban, E., & Langer, T. (2021). Antivirals against the chikungunya virus. In *Viruses*, *13*(9). https://doi.org/10.3390/v13071307

Bhakat, S., Delang, L., Kaptein, S., Neyts, J., Leyssen, P., & Jayaprakash, V. (2015). Reaching beyond HIV/HCV: Nelfinavir as a potential starting point for broad-spectrum protease inhibitors against dengue and chikungunya virus. *RSC Advances*, *5*(104), 85938–85949. https://doi.org/10.1039/c5ra14469h

Bhatia, R., & Narain, J. P. (2009). Re-emerging chikungunya fever: Some lessons from Asia. In *Tropical Medicine and International Health, 14*(8), 940–946. https://doi.org/10.1111/j.1365-3156.2009.02312.x

Breiman, L. (2001). Random Forests. *Machine Learning* 45(1), 5-32.

Burt, F. J., Chen, W., Miner, J. J., Lenschow, D. J., Merits, A., Schnettler, E., Kohl, A., Rudd, P. A., Taylor, A., Herrero, L. J., Zaid, A., Ng, L. F. P., & Mahalingam, S. (2017). Chikungunya virus: an update on the biology and pathogenesis of this emerging pathogen. In *The Lancet Infectious Diseases* 17(4), 107–117. https://doi.org/10.1016/S1473-3099(16)30385-1

Caglioti, C., Lalle, E., Castilletti, C., Carletti, F., Capobianchi, M. R., & Bordi, L. (2013). Chikungunya virus infection: an overview. In *NEW MICROBIOLOGICA 36*. http://www.cdc.gov/chikungunya/map/index.html

Chang, C-C., & Lin, C-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2, 3, Article 27 https://doi.org/10.1145/1961189.1961199*

Chen, M. W., Tan, Y. B., Zheng, J., Zhao, Y., Lim, B. T., Cornvik, T., Lescar, J., Poh Ng, L. F., & Luo, D. (2017). Chikungunya virus nsP4 RNA-dependent RNA polymerase core domain displays detergent-sensitive primer extension and terminal adenylyltransferase activities. *Antiviral Research, 143*, 38-47 https://doi.org/10.1016/j.antiviral.2017.04.001.

Coles, M., Heller, M., & Kessle, H. (2003). NMR-based screening technologies. *DDT* , *8*(17).

da Silva-Júnior, E. F., Leoncini, G. O., Rodrigues, É. E. S., Aquino, T. M., & Araújo-Júnior, J. X. (2017). The medicinal chemistry of Chikungunya virus. In *Bioorganic and Medicinal Chemistry*, *25(16),* 4219–4244. https://doi.org/10.1016/j.bmc.2017.06.049

Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M., & Ahsan, M. J. (2022). Machine Learning in Drug Discovery: A Review. *Artificial Intelligence Review*, *55*(3), 1947–1999. https://doi.org/10.1007/s10462-021-10058-4

de Lamballerie, X., Ninove, L., & Charrel, R. N. (2009). Antiviral Treatment of Chikungunya Virus Infection. In *Infectious Disorders-Drug Targets 9*. http://www.invs.sante.fr/agenda/colloque_chikungunya/

Delang, L., Guerrero, N. S., Tas, A., Quérat, G., Pastorino, B., Froeyen, M., Dallmeier, K., Jochmans, D., Herdewijn, P., Bello, F., Snijder, E. J., de Lamballerie, X., Martina, B., Neyts, J., van Hemert, M. J., & Leyssen, P. (2014). Mutations in the chikungunya virus non-structural proteins cause resistance to favipiravir (T-705), a broad-spectrum antiviral. *Journal of Antimicrobial Chemotherapy*, *69*(10), 2770–2784. https://doi.org/10.1093/jac/dku209

Delang, L., Li, C., Tas, A., Quérat, G., Albulescu, I. C., de Burghgraeve, T., Segura Guerrero, N. A., Gigante, A., Piorkowski, G., Decroly, E., Jochmans, D., Canard, B., Snijder, E. J., Pérez-Pérez, M. J., van Hemert, M. J., Coutard, B., Leyssen, P., & Neyts, J. (2016). The viral capping enzyme nsP1: A novel target for the inhibition of chikungunya virus infection. *Scientific Reports*, *6*. https://doi.org/10.1038/srep31819

Delogu, I., Pastorino, B., Baronti, C., Nougairède, A., Bonnet, E., & de Lamballerie, X. (2011). In vitro antiviral activity of arbidol against Chikungunya virus and characteristics of a selected resistant mutant. *Antiviral Research*, *90*(3), 99–107. https://doi.org/10.1016/j.antiviral.2011.03.182

Dempster, A.P. (1968) A generalization of Bayesian Interference. *J. Royal Stat. Soc. B 30*, 205-247.

Dey, D., Siddiqui, S. I., Mamidi, P., Ghosh, S., Kumar, C. S., Chattopadhyay, S., Ghosh, S., & Banerjee, M. (2019). The effect of amantadine on an ion channel protein from Chikungunya virus. *PLoS Neglected Tropical Diseases*, *13*(7). https://doi.org/10.1371/journal.pntd.0007548

di Masi, J. A., Grabowski, H. G., & Hansen, R. W. (2016). Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, *47*, 20–33. https://doi.org/10.1016/j.jhealeco.2016.01.012

di Mola, A., Peduto, A., la Gatta, A., Delang, L., Pastorino, B., Neyts, J., Leyssen, P., de Rosa, M., & Filosa, R. (2014). Structure-activity relationship study of arbidol derivatives as inhibitors of chikungunya virus replication. *Bioorganic and Medicinal Chemistry*, *22*(21), 6014–6025. https://doi.org/10.1016/j.bmc.2014.09.013

Economopoulou, A., Dominguez, M., Helynck, B., Sissoko, D., Wichmann, O., Quenel, P., Germonneau, P., & Quatresous, I. (2009). Atypical Chikungunya virus infections: Clinical manifestations, mortality and risk factors for severe disease during the 2005-2006 outbreak on Réunion. *Epidemiology and Infection*, *137*(4), 534–541. https://doi.org/10.1017/S0950268808001167

Elbadawi, M., Gaisford, S., & Basit, A. W. (2021). Advanced machine-learning techniques in drug discovery. In *Drug Discovery Today*, 26(3), 769–777. https://doi.org/10.1016/j.drudis.2020.12.003

Enserink Martin. (2007). Infectious diseases. Chikungunya: no longer a third world disease. *Science*, *318*(5858), 1860–1861.

Erin Staples, J., Breiman, R. F., & Powers, A. M. (2009). Chikungunya fever: An epidemiological review of a re-emerging infectious disease. In *Clinical Infectious Diseases*, *49*(6), 942–948. https://doi.org/10.1086/605496

Fang, J., Yang, R., Gao, L., Zhou, D., Yang, S., Liu, A. L., & Du, G. H. (2013). Predictions of buche inhibitors using support vector machine and naive bayesian classification techniques in drug discovery. *Journal of Chemical Information and Modeling*, *53*(11), 3009–3020. https://doi.org/10.1021/ci400331p

Fatma, B., Kumar, R., Singh, V. A., Nehul, S., Sharma, R., Kesari, P., Kuhn, R. J., & Tomar, S. (2020). Alphavirus capsid protease inhibitors as potential antiviral agents for Chikungunya infection. *Antiviral Research*, *179*. https://doi.org/10.1016/j.antiviral.2020.104808

Feibelman, K. M., Fuller, B. P., Li, L., LaBarbera, D. v., & Geiss, B. J. (2018). Identification of small molecule inhibitors of the Chikungunya virus nsP1 RNA capping enzyme. *Antiviral Research*, *154*, 124–131. https://doi.org/10.1016/j.antiviral.2018.03.013

Fernandez-Pol, J.A. & Fernandez-Pol, S. (2010). Method to Control Dengue Viruses in Humans by Picolinic Acid and Derivates. *Thereof 2010. U.S. Patent* 12/175,277.

Foeller, M. E., Nosrat, C., Krystosik, A., Noel, T., Gérardin, P., Cudjoe, N., Mapp-Alexander, V., Mitchell, G., Macpherson, C., Waechter, R., & LaBeaud, A. D.

(2021). Chikungunya infection in pregnancy – reassuring maternal and perinatal outcomes: a retrospective observational study. *BJOG: An International Journal of Obstetrics and Gynaecology*, *128*(6), 1077–1086. https://doi.org/10.1111/1471-0528.16562

Fros, J. J., Liu, W. J., Prow, N. A., Geertsema, C., Ligtenberg, M., Vanlandingham, D. L., Schnettler, E., Vlak, J. M., Shurbier, A., Khromykh, A. A., & Pijlman, G. P. (2010). Chikungunya virus nonstructural protein 2 inhibits type I/II interferon-stimulated JAK-STAT signaling. *J Virol, 84*(20), 10877–10887.

Galán-Huerta, K. A., Rivas-Estilla, A. M., Fernández-Salas, I., Farfan-Ale, J. A., & Ramos-Jiménez, J. (2015). Chikungunya virus: A general overview. *Medicina Universitaria*, *17*(68), 175–183. https://doi.org/10.1016/j.rmu.2015.06.001

Gaulton, A., Hersey, A., Nowotka, M. L., Patricia Bento, A., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrian-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magarinos, M. P., Overington, J. P., Papadatos, G., Smit, I., & Leach, A. R. (2017). The ChEMBL database in 2017. *Nucleic Acids Research*, *45*(D1), D945–D954. https://doi.org/10.1093/nar/gkw1074

Gawriljuk, V. O., Foil, D. H., Puhl, A. C., Zorn, K. M., Lane, T. R., Riabova, O., Makarov, V., Godoy, A. S., Oliva, G., & Ekins, S. (2021). Development of Machine Learning Models and the Discovery of a New Antiviral Compound against Yellow Fever Virus. In *Journal of Chemical Information and Modeling*, 61(8), 3804–3813. American Chemical Society. https://doi.org/10.1021/acs.jcim.1c00460

Gupta, R., Srivastava, D., Sahu, M., Tiwari, S., Ambasta, R. K., & Kumar, P. (2021). Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Molecular Diversity*, *25*(3), 1315–1360. https://doi.org/10.1007/s11030-021-10217-3

Higuera, A., & Ramírez, J. D. (2019). Molecular epidemiology of dengue, yellow fever, Zika and Chikungunya arboviruses: An update. In *Acta Tropica*, *190,* 99–111. https://doi.org/10.1016/j.actatropica.2018.11.010

Ho, Y. J., Liu, F. C., Yeh, C. T., Yang, C. M., Lin, C. C., Lin, T. Y., Hsieh, P. S., Hu, M. K., Gong, Z., & Lu, J. W. (2018). Micafungin is a novel anti-viral agent of chikungunya virus through multiple mechanisms. *Antiviral Research*, *159*, 134–142. https://doi.org/10.1016/j.antiviral.2018.10.005

Hua, C., Lee, R., Hussain, K. M., & Chu, J. J. H. (2019). Macropinocytosis dependent entry of Chikungunya virus into human muscle cells. *PLoS Neglected Tropical Diseases*, *13*(8). https://doi.org/10.1371/journal.pntd.0007610

Jadav, S. S., Sinha, B. N., Hilgenfeld, R., Pastorino, B., de Lamballerie, X., & Jayaprakash, V. (2015). Thiazolidone derivatives as inhibitors of chikungunya virus. *European Journal of Medicinal Chemistry*, *89*, 172–178. https://doi.org/10.1016/j.ejmech.2014.10.042

Jiang, D., Lei, T., Wang, Z., Shen, C., Cao, D., & Hou, T. (2020). ADMET evaluation in drug discovery. 20. Prediction of breast cancer resistance protein inhibition through machine learning. *Journal of Cheminformatics*, *12*(1). https://doi.org/10.1186/s13321-020-00421-y

Kamboj, S., Rajput, A., Rastogi, A., Thakur, A., & Kumar, M. (2022). Targeting non-structural proteins of Hepatitis C virus for predicting repurposed drugs using QSAR and machine learning approaches. *Computational and Structural Biotechnology Journal*, *20*, 3422–3438. https://doi.org/10.1016/j.csbj.2022.06.060

Kaur, P., Thiruchelvan, M., Lee, R. C. H., Chen, H., Chen, K. C., Ng, M. L., & Chu, J. J. H. (2013). Inhibition of Chikungunya virus replication by harringtonine, a novel antiviral that suppresses viral protein expression. *Antimicrobial Agents and Chemotherapy*, *57*(1), 155–167. https://doi.org/10.1128/AAC.01467-12

Khan, M., Santhosh, S. R., Tiwari, M., Lakshmana Rao, P. v., & Parida, M. (2010). Assessment of in vitro prophylactic and therapeutic efficacy of chloroquine against Chikungunya virus in Vero cells. *Journal of Medical Virology*, *82*(5), 817–824. https://doi.org/10.1002/jmv.21663

Khongwichit, S., Chansaenroj, J., Chirathaworn, C., & Poovorawan, Y. (2021). Chikungunya virus infection: molecular biology, clinical characteristics, and epidemiology in Asian countries. In *Journal of Biomedical Science, 28*(1). https://doi.org/10.1186/s12929-021-00778-8

Kovacikova, K., & van Hemert, M. J. (2020). Small-Molecule Inhibitors of Chikungunya Virus: Mechanisms of Action and Antiviral Drug Resistance. In *Antimicrobial Agents and Chemotherapy, 64*(12). https://doi.org/10.1128/AAC.01788-20

LaBeaud, A. D., Bashir, F., & King, C. H. (2011). Measuring the burden of arboviral diseases: The spectrum of morbidity and mortality from four prevalent infections. *Population Health Metrics*, *9*. https://doi.org/10.1186/1478-7954-9-1

Lantz, B. (2015). Machine learning with R : discover how to build machine learning algorithms, prepare data, and dig deep into data prediction techniques with R. *Packt Publishing*. *Second edition.*

Lo Presti, A., Ciccozzi, M., Cella, E., Lai, A., Simonetti, F. R., Galli, M., Zehender, G., & Rezza, G. (2012). Origin, evolution, and phylogeography of recent epidemic CHIKV strains. *Infect Genet Evol, 12*, 392-398.

Malet, H., Coutard, B., Jamal, S., Dutartre, H., Papageorgiou, N., Neuvonen, M., Ahola, T., Forrester, N., Gould, E. A., Lafitte, D., Ferron, F., Lescar, J., Gorbalenya, A. E., de Lamballerie, X., & Canard, B. (2009). The Crystal Structures of Chikungunya and Venezuelan Equine Encephalitis Virus nsP3 Macro Domains Define a Conserved Adenosine Binding Pocket. *Journal of Virology*, *83*(13), 6534–6545. https://doi.org/10.1128/jvi.00189-09

Matsumoto, A., Aoki, S., & Ohwada, H. (2016). Comparison of Random Forest and SVM for Raw Data in Drug Discovery: Prediction of Radiation Protection and Toxicity Case Study. *International Journal of Machine Learning and Computing*, *6*(2), 145–148. https://doi.org/10.18178/ijmlc.2016.6.2.589

Meyes, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., & Lin, C.-C. (2022). Probability Theory Group (Formerly: E1071),.*Misc Functions of the Department of Statistics, TU Wien* (Version 1.7-12).

Moizéis, R. N. C., Fernandes, T. A. A. de M., Guedes, P. M. da M., Pereira, H. W. B., Lanza, D. C. F., Azevedo, J. W. V. de, Galvão, J. M. de A., & Fernandes, J. V. (2018). Chikungunya fever: a threat to global public health. In *Pathogens and Global Health*, 112(14), 182–194. https://doi.org/10.1080/20477724.2018.1478777

Mourad, O., Makhani, L., & Chen, L. H. (2022). Chikungunya: An Emerging Public Health Concern. In *Current Infectious Disease Reports*. Springer. https://doi.org/10.1007/s11908-022-00789-y

Nguyen, P. T. V., Yu, H., & Keller, P. A. (2014). Discovery of in silico hits targeting the nsP3 macro domain of chikungunya virus. *Journal of Molecular Modeling*, *20*(5). https://doi.org/10.1007/s00894-014-2216-6

Panas, M. D., Ahola, T., & McInerney, G. M. (2014). The C-terminal repeat domains of nsP3 from the old-world alphaviruses bind directly to G3BP. *J Virol., 88*(10), 5888–5893

Pérez-Pérez, M. J., Delang, L., Ng, L. F. P., & Priego, E. M. (2019). Chikungunya virus drug discovery: still a long way to go? In *Expert Opinion on Drug Discovery*, 14(9), 855–866. https://doi.org/10.1080/17460441.2019.1629413

Pohjala, L., Utt, A., Varjak, M., Lulla, A., Merits, A., Ahola, T., & Tammela, P. (2011). Inhibitors of alphavirus entry and replication identified with a stable Chikungunya replicon cell line and virus-based assays. *PLoS ONE*, *6*(12). https://doi.org/10.1371/journal.pone.0028923

Powers, A. M. (2018). Vaccine and therapeutic options to control chikungunya virus. In *Clinical Microbiology Reviews*, *31*(1). American Society for Microbiology. https://doi.org/10.1128/CMR.00104-16

Quinlan R (1993). C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*, http://www.rulequest.com/see5-unix.html [24/11/2022].

Rezza, G., Nicoletti, L., Angelini, R., Romi, R., Finarelli, A. C., Panning, M., Cardioli, P., Fortuna, C., Boros, S., MAgurano, F., Silvi, G., Angelini, P., Dottori, M., Ciufolini, M. G., Majori, G. C., Cassone, A. (2007). Infection with chikungunya virus in Italy: an outbreak in a temperate region. *Lancet*, *370*(9602), 1840–1846.

Ross RW. (1956). The Newala epidemic. III. The virus: isolation, pathogenic properties and relationship to the epidemic. *J Hyg (Lond)*, *54*(2), 177–191.

Robinson, M. C. (1955). Clinical Features. In *communications an epidemic of virus disease in southern province, Tanganyika territory, 49*(1).

Santana, A. C., Silva Filho, R. C., Menezes, J. C. J. M. D. S., Allonso, D., & Campos, V. R. (2021). Nitrogen-based heterocyclic compounds: A promising class of antiviral agents against chikungunya virus. In *Life*, *11*(1). https://doi.org/10.3390/life11010016

Schwartz, O., & Albert, M. L. (2010). Biology and pathogenesis of chikungunya virus. In *Nature Reviews Microbiology,* *8*(7), 491–500. https://doi.org/10.1038/nrmicro2368

Sheridan, R. P., Wang, W. M., Liaw, A., Ma, J., & Gifford, E. M. (2016). Extreme Gradient Boosting as a Method for Quantitative Structure-Activity Relationships. *Journal of Chemical Information and Modeling*, *56*(12), 2353–2360. https://doi.org/10.1021/acs.jcim.6b00591

Silva, L. A., & Dermody, T. S. (2017). Chikungunya virus: Epidemiology, replication, disease mechanisms, and prospective intervention strategies. In *Journal of Clinical Investigation,* *127*(3), 737–749. https://doi.org/10.1172/JCI84417

Simon, F., Javelle, E., Oliver, M., Leparc-Goffart, I., & Marimoutou, C. (2011). Chikungunya virus infection. *Current Infectious Disease Reports*, *13*(3), 218–228. https://doi.org/10.1007/s11908-011-0180-1

Singh, H., Mudgal, R., Narwal, M., Kaur, R., Singh, V. A., Malik, A., Chaudhary, M., & Tomar, S. (2018). Chikungunya virus inhibition by peptidomimetic inhibitors targeting virus-specific cysteine protease. *Biochimie*, *149*, 51–61. https://doi.org/10.1016/j.biochi.2018.04.004

Soumahoro, M. K., Gérardin, P., Boëlle, P. Y., Perrau, J., Fianu, A., Pouchot, J., Malvy, D., Flahault, A., Favier, F., & Hanslik, T. (2009). Impact of Chikungunya virus infection on health status and quality of life: A retrospective cohort study. *PLoS ONE*, *4*(11). https://doi.org/10.1371/journal.pone.0007800

Suhrbier, A. (2019). Rheumatic manifestations of chikungunya: emerging concepts and interventions. In *Nature Reviews Rheumatology*, *15*(10), 597–611. https://doi.org/10.1038/s41584-019-0276-9

Tolle, M. A. (2009). Mosquito-borne Diseases. *Current Problems in Pediatric and Adolescent Health Care*, *39*(4), 97–140. https://doi.org/10.1016/j.cppeds.2009.01.001

Tripathi, P. K., Soni, A., Singh Yadav, S. P., Kumar, A., Gaurav, N., Raghavendhar, S., Sharma, P., Sunil, S., Ashish, Jayaram, B., & Patel, A. K.

(2020). Evaluation of novobiocin and telmisartan for anti-CHIKV activity. *Virology*, *548*, 250–260. https://doi.org/10.1016/j.virol.2020.05.010

Tsao, S. C. H., Wang, J., Wang, Y., Behren, A., Cebon, J., & Trau, M. (2018). Characterising the phenotypic evolution of circulating tumour cells during treatment. *Nature Communications*, *9*(1). https://doi.org/10.1038/s41467-018-03725-8

Urakova, N., Kuznetsova, V., Crossman, D. K., Sokratian, A., Guthrie, D. B., Kolykhalov, A. A., Lockwood, M. A., Natchus, M. G., Crowley, M. R., Painter, G. R., Frolova, E. I., & Frolov, I. (2018). β- d - N 4 -Hydroxycytidine Is a Potent Anti-alphavirus Compound That Induces a High Level of Mutations in the Viral Genome. *Journal of Virology*, *92*(3). https://doi.org/10.1128/jvi.01965-17

Utt, A., Das, P.K., Varjak, M., Lulla, V., Lulla, A. & Merits, A. (2015). Mutations conferring a noncytotoxic phenotype on chikungunya virus replicons compromise enzymatic properties of nonstructural protein 2. *J Virol.*, *89*(6), 3145–3162

Vairo, F., Haider, N., Kock, R., Ntoumi, F., Ippolito, G., & Zumla, A. (2019). Chikungunya: Epidemiology, Pathogenesis, Clinical Features, Management, and Prevention. In *Infectious Disease Clinics of North America, 33*(4), 1003–1025. https://doi.org/10.1016/j.idc.2019.08.006

Valdés López, J. F., Velilla, P. A., & Urcuqui-Inchima, S. (2019). Chikungunya Virus and Zika Virus, Two Different Viruses Examined with a Common Aim: Role of Pattern Recognition Receptors on the Inflammatory Response. In *Journal of Interferon and Cytokine Research, 39*(9), 507–521. https://doi.org/10.1089/jir.2019.0058

Vapnik, V. N. (1999). The Nature of Statistical Learning Theory. *Springer. ISBN: 0387987800*

Varghese, F. S., Rausalu, K., Hakanen, M., Saul, S., Kümmerer, B. M., Susi, P., Merits, A., & Ahola, T. (2017). Obatoclax inhibits alphavirus membrane fusion by neutralizing the acidic environment of endocytic compartments. *Antimicrobial Agents and Chemotherapy*, *61*(3). https://doi.org/10.1128/AAC.02227-16

Venables, W. N. and Ripley, B. D. (2010) Modern Applied Statistics with S. *Springer New York, NY, 4*(11), 498.

Wada, Y., Orba, Y., Sasaki, M., Kobayashi, S., Carr, M. J., Nobori, H., Sato, A., Hall, W. W., & Sawa, H. (2017). Discovery of a novel antiviral agent targeting the nonstructural protein 4 (nsP4) of chikungunya virus. *Virology*, *505*, 102–112. https://doi.org/10.1016/j.virol.2017.02.014

Wale, N. (2011). Machine learning in drug discovery and development. In *Drug Development Research, 72*(1), 112–119. https://doi.org/10.1002/ddr.20407

Wang, L., Wang, M., Yan, A., Dai, B. (2013). Using self-organizing map (SOM) and support vector machine (SVM) for classification of selectivity of ACAT inhinitors. *Mol. Diversity 2013, 17,* 85-96

Wang, Y. M., Lu, J. W., Lin, C. C., Chin, Y. F., Wu, T. Y., Lin, L. I., Lai, Z. Z., Kuo, S. C., & Ho, Y. J. (2016). Antiviral activities of niclosamide and nitazoxanide against chikungunya virus entry and transmission. *Antiviral Research*, *135*, 81–90. https://doi.org/10.1016/j.antiviral.2016.10.003

Weber, C., Sliva, K., von Rhein, C., Kümmerer, B. M., & Schnierle, B. S. (2015). The green tea catechin, epigallocatechin gallate inhibits chikungunya virus infection. *Antiviral Research*, *113*, 1–3. https://doi.org/10.1016/j.antiviral.2014.11.001

White, G.B. (2004). Medical acarology and entomology: mosquitoes. *Manson's Tropical Diseases (21st Edition)*, *Elsevier*, London, England.

https://www.aemps.gob.es/la-aemps/ultima-informacion-de-la-aemps-acerca-del-covid-19/vacunas-contra-la-covid-19/desarrollo-de-vacunas/ [22/11/2022]

https://www.clinicaltrials.gov/ct2/results?cond=Chikungunya&age_v=&gndr=&type=&rslt=&phase=2&Search=Apply [22/11/2022]

https://valneva.com/press-release/valneva-successfully-completes-pivotal-phase-3-trial-of-single-shot-chikungunya-vaccine-candidate/ [22/11/2022]

https://www.clinicaltrials.gov/ct2/show/results/NCT04546724?cond=Chikungunya&phase=2&draw=2&rank=6 [22/11/2022]

https://www.drugs.com/drug-class/phenothiazine-antipsychotics.html [22/11/2022]

https://xgboost.readthedocs.io/en/stable/R-package/index.html [01/12/2022]

https://towardsdatascience.com/xgboost-extreme-gradient-boosting-how-to-improve-on-regular-gradient-boosting-5c6acf66c70a [01/12/2022]

Anexos

Descripción de descriptores

Smiles: (simplified molecular input line entry system), es una notación química que permite al usuario representar una estructura química de una manera que pueda ser utilizada por el ordenador.

Standard type: half maximal effective concentraction, concentración mínima requerida de un fármaco, anticuerpo o sustancia tóxica para obtener un efecto del 50%.

Standard value: concentración nanomolecular donde la molécula produce el 50% del efecto máximo.

Target ChEMBL ID: diana sobre la que actúan las diferentes moléculas.

Activity: Si las moléculas obtenidas presentan o no bioactividad frente a la molécula de estudio. 1 presenta actividada 0 no presenta

Energy: energía de la disposición actual por coordenadas

SlogP: es el logaritmo del coeficiente de reparto octanol/agua (incluidos los hidrógenos implícitos. Esta propiedad es un modelo de contribución atómica que calcula el logP a partir de la estructura dada, es decir, el estado de protonación correcto (Wildman & Crippen, 1999)

SMR: es la refractividad molecular incluyendo los hidrógenos implícitos. También asume un estado de protonación correcto. (Wildman & Crippen, 1999)

TPSA: la superficie polar topológica es la suma de la superficie de todos los átomos o moléculas polares, incluidos también sus átomos de hidrógeno unidos. (Ertl et al., 2000)

AMW: peso molecular promedio (Shimizu et al., 2021)

MW: peso molecular exacto.

HBALipinski y HBDLipinski: receptores y donadores de puentes de hidrógeno respectivamente calculados según las reglas de Lipinski (Lipinski et al., 1997)

NumRotatableBonds: número de enlaces giratorios

NumHBD: número de donadores de puentes de hidrógeno

NumHBA: número de receptores de puentes de hidrógeno

NumAmideBonds: número de enlaces amida

NumHeteroAtoms: número de heteroátomos

NumHeavyAtoms: número de átomos pesados

NumAtoms: número de átomos

NumStereocenters: número de estereocentros

NumUnspecifiedStereocenters: número de estereocentros inespecíficos

NumRings: número de anillos

NumAromaticRings: número de anillos aromáticos

NumSaturatedRings: número de anillos saturados

NumAliphaticRings: número de anillos alifáticos

NumAromaticHeterocycles: número de compuestos heterocíclicos aromáticos

NumSaturatedHeterocycles: número de compuestos heterocíclicos saturados

NumAliphaticHeterocycles: número de compuestos heterocíclicos alifáticos

NumAromaticCarbocycles: número de compuestos carbociclo aromático

NumSaturatedCarbocycles: número de compuestos carbociclo saturado

NumAliphaticCarbocycles: número de compuestos carbociclo alifáticos.

FractionCSP3: la relación del número de carbonos hibridados sobre el recuento total de carbonos de cada molécula (Daina et al., 2017)

Chiv/Chin: es un conteo ponderado de un tipo dado de subgrafo, donde existen dos categorías: el orden (número de aristas) y el tipo (disposición particular de los bordes) (Hall' & Kier, 1991)

HallKierAlpha: comparan grafos moleculares con grafos moleculares máximos/mínimos intentando capturar los diferentes aspectos de la estructura molecular (Hall' & Kier, 1991)

Kappa: los índices kappa derivan de recuentos de 1-enlace, 2-enlace y 3-enlace a fragmentos, donde cada conteo relativo a un fragmento en estructuras referencia, posee un valor máximo y mínimo para ese número de átomos. (Nandi & Bagchi, 2012)

Peoe: método de ecualización parcial de electronegatividades orbitales que sirve para calcular las cargas parciales atómicas (Gasteiger & Marsili, 1980)

MQN: números cuánticos moleculares son un recuento de las características y estructurales simples como tipo de átomos, enlaces, polaridad y topología, siendo así capaces de analizar grandes bases de datos moleculares (Nguyen et al., 2009)

BIBLIOGRAFÍA

Nguyen, K. T., Blum, L. C., van Deursen, R., & Reymond, J. L. (2009). Classification of organic molecules by molecular quantum numbers. *ChemMedChem*, *4*(11), 1803–1805. https://doi.org/10.1002/cmdc.200900317

Gasteiger, J., & Marsili, M. (1980). Iterative partial equalization of orbital electronegativity-a rapid access to atomic charges.*Tetrahedron, 36, 3219-3288.*

Nandi, S., & Bagchi, M. C. (2012). Importance of Kier-Hall Topological Indices in the QSAR of Anticancer Drug Design. In *Current Computer-Aided Drug Design*, *8*.

Hall', L. H., & Kiert, L. B. (1991). The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling. *Reviews in Computational Chemistry,2.*

Daina, A., Michielin, O., & Zoete, V. (2017). SwissADME: A free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports*, *7*. https://doi.org/10.1038/srep42717

Lipinski, C. A., Dominy, B. W., & Feeney, P. J. (1997). drug delivery reviews Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. In *Advanced Drug Delivery Reviews*, *23*.

Wildman, S. A., & Crippen, G. M. (1999). Prediction of physicochemical parameters by atomic contributions. *Journal of Chemical Information and Computer Sciences*, *39*(5), 868–873. https://doi.org/10.1021/ci990307l

Ertl, P., Rohde, B., & Selzer, P. (2000). Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *Journal of Medicinal Chemistry*, *43*(20), 3714–3717. https://doi.org/10.1021/jm000942e

Labute, P. (2000). A widely applicable set of descriptors. *Chemical Computing Group Inc., Journal of Molecular Graphics and Modelling 18,466-477.*

Shimizu, Y., Sasaki, T., Takeshita, J. I., Watanabe, M., Shizu, R., Hosaka, T., & Yoshinari, K. (2021). Identification of average molecular weight (AMW) as a useful chemical descriptor to discriminate liver injury-inducing drugs. *PLoS ONE*, *16*(6 June). https://doi.org/10.1371/journal.pone.0253855

Obtención de los descriptores

```
descriptor_names = list(rdMolDescriptors.Properties.GetAvailableProperties())
get_descriptors = rdMolDescriptors.Properties(descriptor_names)
def smi_to_descriptors(smile):
    mol = Chem.MolFromSmiles(smile)
    descriptors = []
    if mol:
        descriptors = np.array
```

```
library(readxl)
library(dplyr)
```

```
data_<-"CHEMBL4296563_Chikungunya_Desc.xlsx"
data<-read_xlsx(data_)
head(data)
```

Por lo tanto tenemos un total de 129 variables (128 predictoras y una a predecir que es *Activity*). El resto de nombres de variables es:

Al comienzo existen varias columnas que no son necesarias para el entrenamiento y, por tanto, se eliminan. Estas son el ID de la molécula, el ID de la molécula sobre la que actúa que siempre es el **Chikungunya virus**, el valor estandar y sus unidades.

```
Data<-select(data,-"Molecule ChEMBL ID", -"Standard Type", -"Standard Value",
             -"Standard Units", -"Target ChEMBL ID")
```

Por el momento también se quitan las dos variables de *Smiles* para poder emplear los datos.

```
Data<-select(Data,-"Smiles (Canonical)", -"Smiles", -"Standard Relation")
```

```
summary(Data)
```

Se puede observar que en un gran número de las variables existen columnas con valor 0 por lo tanto esta propiedad no está indicando nada y puede ser eliminada

```
Data<-select(Data,-"slogp_VSA9",-"smr_VSA8",-"MQN7",-"MQN18",-"MQN22",-"MQN23",-"MQN38",
             -"MQN39")
```

```
ncol(Data)
```

```
round(cor(Data),3)
```

No se ha obtenido ninguna variable con una correlación mayor o igual de +- 0.8 respecto a la variable a predecir por lo que se sigue empleando el resto de variables predictoras.

Aunque parece que si que puede existir correlaciones entre sí, por lo que es muy posible que exista multicolinealidad entre ellas. Se eliminan aquellas variables que presenta alta correlación entre sí, ya que podrían afectar al modelo.

```
Data<-select(Data,-"SMR",-"LabuteASA",-"TPSA",-"AMW",-"ExactMW",-"NumLipinskiHBA",
             -"NumLipinskiHBD",-"NumRotatableBonds",-"NumHBA",-"NumHeteroAtoms",
             -"NumHeavyAtoms",-"NumAtoms",-"NumStereocenters",-"NumAliphaticRings",
             -"Chi0v",-"Chi1v",-"Chi2v",-"Chi3v",-"Chi4v",-"Chi1n",-"Chi2n",-"Chi3n",
             -"kappa1",-"kappa2",-"kappa3",-"slogp_VSA2",-"slogp_VSA3",-"slogp_VSA4",
             -"slogp_VSA5",-"slogp_VSA6",-"slogp_VSA11",-"smr_VSA1",-"smr_VSA5",
             -"peoe_VSA1",-"peoe_VSA9",-"peoe_VSA14",-"MQN1",-"MQN10",-"MQN11",
             -"MQN12",-"MQN13",-"MQN16",-"MQN17",-"MQN19",-"MQN20",-"MQN24",-"MQN42")
```

Se mide la varianza de las variables y se eliminan aquellas que presentan poca variación

```
sapply(Data, FUN=var)
```

```
Data<-select(Data,-"FractionCSP3",-"MQN3",-"MQN4",-"MQN5",-"MQN25",-"MQN34")
```

```
ncol(Data)
```

Se modifica la variable *Activity* a tipo factorial ya que está formado por valores 0 y 1 que representan si existe (1) o no existe (0) actividad de la molécula empleada sobre el target.

Se normalizan los datos

```
# Se crea la función de normalización
normalize<-function(x){
  return((x-min(x))/(max(x)-min(x)))
}

# Se aplica a las diferentes variables
data_norm<-lapply(Data[2:60],normalize)
data_norm<-as.data.frame(data_norm)
```

```
summary(data_norm)
```

```
data_norm<-bind_cols(Data$Activity,data_norm)
colnames(data_norm)[1]<-"Activity"
str(data_norm)
```

Se pasa la variable *Activity* de tipo numérico a tipo factor.

```
data_norm$Activity<-factor(data_norm$Activity)
str(data_norm$Activity)
```

Una vez se han normalizado los datos se vuelven a añadir las columnas de *Smiles* y *Smiles (Canonical)*.

```
data_norm<-bind_cols(data$Smiles,data_norm)
colnames(data_norm)[1]<-"Smiles"
```

```
data_norm<-bind_cols(data$`Smiles (Canonical)`,data_norm)
colnames(data_norm)[1]<-"Smiles (Canonical)"
```

Ya se puede comenzar a trabajar con estos datos

```
data<-read_xlsx("CHEMBL4296563_Chikungunya_Desc_NormDimRed_7000nm.xlsx")
```

```
Data$Activity<-factor(Data$Activity)
Data_<-data.frame(Data)
```

Selección aleatoria de los datos

```
set.seed(12345)
training<-sample(272,floor(272*0.670))
data_train<-Data_[training,-1]
data_test<-Data_[-training,-1]

data_labels<-Data_[1]
set.seed(12345)
labels<-sample(272,floor(272*0.670))
labels_train<-data_labels[labels,]
labels_test<-data_labels[-labels,]
prop.table(table(labels_train))
prop.table(table(labels_test))
```

```
summary(labels_test)
summary(labels_train)
```

Las muestras de actividad y no actividad se encuentran distribuidas equitativamente por lo que se puede continuar.

k-Nearest Neighbour

El primer algoritmo de clasificación a emplear será *k-Nearest Neighbour* donde se explorarán los valores para el número de vecinos **k**=1,3,5,7 y 11.

Modelo k=1

```
#Predicción para k = 1
set.seed(12345)
test_pred1<-knn(train = data_train, test = data_test, cl=labels_train, k=1, prob=TRUE)
res<-table(test_pred1,labels_test)
confusionMatrix(res, positive="1")
```

```
prob<-attr(test_pred1,"prob")
prob<-ifelse(test_pred1 =="1",prob,1-prob)
ROC_knn<- roc(labels_test, prob)
ROC_knn_auc<-auc(ROC_knn)
ROC_knn_auc
```

Modelo k=3

```r
#Predicción para k = 3
set.seed(12345)
test_pred3<-knn(train = data_train, test = data_test, cl=labels_train, k=3)
res<-table(test_pred3,labels_test)
confusionMatrix(res, positive="1")
```

Modelo k=5

```r
#Predicción para k = 5
set.seed(12345)
test_pred5<-knn(train = data_train, test = data_test, cl=labels_train, k=5)
res<-table(test_pred5,labels_test)
confusionMatrix(res, positive="1")
```

Modelo k=7

```r
#Predicción para k = 7
set.seed(12345)
test_pred7<-knn(train = data_train, test = data_test, cl=labels_train, k=7)
res<-table(test_pred7,labels_test)
confusionMatrix(res, positive="1")
```

Modelo k=11

```r
#Predicción para k = 11
set.seed(12345)
test_pred11<-knn(train = data_train, test = data_test, cl=labels_train, k=11)
res<-table(test_pred11,labels_test)
confusionMatrix(res, positive="1")
```

```r
set.seed(12345)
k<-c(1,3,5,7,11)
for(i in k){
  test_pred<-knn(train = data_train, test = data_test, cl=labels_train,    k=i, prob=TRUE)
  prob<-attr(test_pred,"prob")
  prob1<-ifelse(test_pred =="1",prob,1-prob)
  auc<-auc(labels_test,prob1)
  pred_knn<- ROCR::prediction(prob1, labels_test)
  pred_knn<-performance(pred_knn,"tpr","fpr")
  plot(pred_knn, avg="threshold",col="red",lwd=2,
       main=paste("Curva ROC para k=",i,"Valor AUC=", round(auc,4)))
}
```

Naive Bayes

Laplace = 0

Para el caso en que laplace es igual a 0:

```
set.seed(12345)
data_bayes<-naiveBayes(data_train, labels_train, laplace = 1)
bayes_pred<-predict(data_bayes, data_test)
confusionMatrix(bayes_pred,labels_test, positive = "1")
```

Se obtienen 36 verdaderos negativos, 22 verdaderos positivos, 11 falsos positivos y 21 falsos negativos. Con una precisión de 0.6444, con valor aceptable 0.2804 y con una sensibilidad y especificidad de 0.5116 y 0.7660 respectivamente.

Laplace = 1

Para laplace = 1

```
set.seed(12345)
data_bayes2<-naiveBayes(data_train, labels_train, laplace = 1)
bayes_pred2<-predict(data_bayes2, data_test)
confusionMatrix(bayes_pred2,labels_test, positive = "1")
```

Mismo resultado

Laplace 5

```
set.seed(12345)
data_bayes5<-naiveBayes(data_train, labels_train, laplace = 5)
bayes_pred5<-predict(data_bayes5, data_test)
confusionMatrix(bayes_pred5,labels_test, positive = "1")
```

```
pred_nb<-predict(data_bayes, data_test, type="raw")
ROC_nb<-roc(labels_test, pred_nb[,2])
ROC_nb_auc<-auc(ROC_nb)
ROC_nb_auc
```

SVM

Implementación con el modelo para dos tipos de *kernel* una para un modelo de tipo lineal (vanilladot) y otra para un kernel Gaussiano (rbfdot).

```
set.seed(12345)
training<-sample(272,floor(272*0.670))
vector_train<-Data_[training,]
vector_test<-Data_[-training,]
```

Kernel lineal

Modelo kernel de función lineal C=1

```
set.seed(12345)
vector_lineal<-ksvm(Activity~., data = vector_train, kernel = "vanilladot",
                    scale=FALSE,prob.model=TRUE)
```

```
vector_lineal
```

Para un coste 1 se obtiene una tasa de erro de 0.1703 donde 1 de cada 6 predicciones podría ser errónea

```
vector_lineal_pred<-predict(vector_lineal, vector_test)
agreetment<-vector_lineal_pred==vector_test$Activity
prop.table(table(agreetment))
```

```
repre_lineal<-table(vector_lineal_pred,vector_test$Activity)
cmat_lineal<-confusionMatrix(repre_lineal, positive ="1")
cmat_lineal
```

Precisión ligeramente superior a la del modelo anterior

```
vector_lineal_score<-predict(vector_lineal,vector_test, type="probabilities")[,2]
ROC_svm<-roc(vector_test$Activity, vector_lineal_score)
ROC_svm_auc<-auc(ROC_svm)
ROC_svm_auc
```

```
vector_lineal_score<-predict(vector_lineal,vector_test, type="probabilities")[,2]
pred_lineal<- ROCR::prediction(vector_lineal_score, vector_test$Activity)
perf_lineal<-performance(pred_lineal,"tpr","fpr")
plot(perf_lineal, lwd=2,colorize=TRUE, main="ROC: Actuación SVM kernel linear")
```

Modelo kernel de función lineal C=3

```
set.seed(12345)
vector_lineal3<-ksvm(Activity~., data = vector_train, kernel = "vanilladot",C=3,
                     scale=FALSE,prob.model=TRUE)
```

```
vector_lineal3
```

Para un coste 1 se obtiene una tasa de erro de 0.1648 donde 1 de cada 6 predicciones podría ser errónea

```
vector_lineal_pred3<-predict(vector_lineal3, vector_test)
agreetment<-vector_lineal_pred3==vector_test$Activity
prop.table(table(agreetment))
```

```
repre_lineal3<-table(vector_lineal_pred3,vector_test$Activity)
cmat_lineal3<-confusionMatrix(repre_lineal3, positive ="1")
cmat_lineal3
```

Modelo kernel de función lineal C=7

```
set.seed(12345)
vector_lineal7<-ksvm(Activity~., data = vector_train, kernel = "vanilladot",C=7,
                     scale=FALSE,prob.model=TRUE)
```

```
vector_lineal7
```

Para un coste 1 se obtiene una tasa de erro de 0.1648 donde 1 de cada 6 predicciones podría ser errónea

```
vector_lineal_pred7<-predict(vector_lineal7, vector_test)
agreetment<-vector_lineal_pred7==vector_test$Activity
prop.table(table(agreetment))
```

```
repre_lineal7<-table(vector_lineal_pred7,vector_test$Activity)
cmat_lineal7<-confusionMatrix(repre_lineal7,positive ="1")
cmat_lineal7
```

Kernel rbfdot

Modelo con kernel Gaussiano RBF.

```
set.seed(12345)
vector_gauss<-ksvm(Activity~., data = vector_train, kernel = "rbfdot",
                   scale = FALSE, prob.model=TRUE)
vector_gauss
```

Error de 0.1923 ligeramente superior con respecto al lineal.

```
vector_gauss_pred<-predict(vector_gauss, vector_test)
agreetment_gauss<-vector_gauss_pred==vector_test$Activity
prop.table(table(agreetment_gauss))
```

El número de predicciones erróneas es mayor que en el caso anterior.

```
repre_gauss<-table(vector_gauss_pred,vector_test$Activity)
cmat_gauss<-confusionMatrix(repre_gauss,positive ="1")
cmat_gauss
```

La precisión, así como la sensibilidad y la especificidad bajan ligeramente respecto al modelo SVM lineal.

```
vector_gauss_score<-predict(vector_gauss,vector_test, type="probabilities")[,2]
pred_radial<- ROCR::prediction(vector_gauss_score, vector_test$Activity)
perf_radial<-performance(pred_radial,"tpr","fpr")
plot(perf_radial, lwd=2,
     colorize=TRUE, main="ROC: Actuación SVM kernel radial")
```

Modelo con kernel Gaussiano RBF C=3

```
set.seed(12345)
vector_gauss3<-ksvm(Activity~., data = vector_train, kernel = "rbfdot", C=3,
                    scale = FALSE, prob.model=TRUE)
vector_gauss3
```

```
vector_gauss_pred3<-predict(vector_gauss3, vector_test)
agreetment_gauss3<-vector_gauss_pred3==vector_test$Activity
prop.table(table(agreetment_gauss3))
```

```
repre_gauss3<-table(vector_gauss_pred3,vector_test$Activity)
cmat_gauss3<-confusionMatrix(repre_gauss3,positive ="1")
cmat_gauss3
```

Modelo con kernel Gaussiano RBF C=7

```
set.seed(12345)
vector_gauss7<-ksvm(Activity~., data = vector_train, kernel = "rbfdot", C=7,
                    scale = FALSE, prob.model=TRUE)
vector_gauss7
```

```
vector_gauss_pred7<-predict(vector_gauss7, vector_test)
agreetment_gauss7<-vector_gauss_pred7==vector_test$Activity
prop.table(table(agreetment_gauss7))
```

```
repre_gauss7<-table(vector_gauss_pred7,vector_test$Activity)
cmat_gauss7<-confusionMatrix(repre_gauss7,positive ="1")
cmat_gauss7
```

XGBoost

```
library(xgboost)
```

```
set.seed(12345)
xgb_labels<-as.vector(labels_train)
xgb_train<-as.matrix(data_train)
```

nrounds = 1

max.depth 12

```
bst<-xgboost(data=xgb_train, label = xgb_labels, max.depth = 12, eta = 1, nthread = 2,
             nrounds = 1, objective = "binary:logistic", verbose = 2)
```

```
pred<-predict(bst, as.matrix(data_test))
```

```
xgb_prediction<-as.numeric(pred>0.5)
xgb_prediction<-as.factor(xgb_prediction)
```

```
err<-mean(as.numeric(pred>0.5) !=labels_test)
print(paste("test-error=", err))
```

```
xgb_tree<-confusionMatrix(xgb_prediction,labels_test, positive ="1")
xgb_tree
```

max.depth 6

```
bst1<-xgboost(data=as.matrix(data_train), label = xgb_labels, max.depth = 6, eta = 1,
              nthread = 2, nrounds = 1,
              objective = "binary:logistic", verbose = 2)
```

```
pred1<-predict(bst1, as.matrix(data_test))
```

```
xgb_prediction1<-as.numeric(pred1>0.5)
xgb_prediction1<-as.factor(xgb_prediction1)
```

```
xgb_tree1<-confusionMatrix(xgb_prediction1,labels_test, positive ="1")
xgb_tree1
```

nrounds = 6

max.depth 12

```
bst3<-xgboost(data=as.matrix(data_train), label = xgb_labels, max.depth = 12, eta = 1,
              nthread = 2, nrounds = 6, objective = "binary:logistic",
              verbose = 2)
```

```
pred3<-predict(bst3, as.matrix(data_test))
```

```
xgb_prediction3<-as.numeric(pred>0.5)
xgb_prediction3<-as.factor(xgb_prediction3)
```

```
xgb_tree3<-confusionMatrix(xgb_prediction3,labels_test, positive ="1")
xgb_tree3
```

max.depth 6

```
bst4<-xgboost(data=as.matrix(data_train), label = xgb_labels, max.depth = 6,
              eta = 1, nthread = 2, nrounds = 6, objective = "binary:logistic",
              verbose = 2)
```

```
pred4<-predict(bst4, as.matrix(data_test))
```

```
xgb_prediction4<-as.numeric(pred4>0.5)
xgb_prediction4<-as.factor(xgb_prediction4)
```

```
xgb_tree4<-confusionMatrix(xgb_prediction4,labels_test, positive ="1")
xgb_tree4
```

```
pred_xgb<-predict(bst4, as.matrix(data_test), type="prob")
ROC_xgb<-roc(labels_test, pred_xgb)
ROC_xgb_auc<-auc(ROC_xgb)
ROC_xgb_auc
```

nrounds 12

max.depth 12

```
bst5<-xgboost(data=as.matrix(data_train), label = xgb_labels, max.depth = 12,eta = 1,
              nthread = 2, nrounds = 12, objective = "binary:logistic", verbose = 2)
```

```
pred5<-predict(bst5, as.matrix(data_test))
```

```
xgb_prediction5<-as.numeric(pred5>0.5)
xgb_prediction5<-as.factor(xgb_prediction5)
```

```
xgb_tree5<-confusionMatrix(xgb_prediction5,labels_test, positive ="1")
xgb_tree5
```

max.depth 6

```
bst6<-xgboost(data=as.matrix(data_train), label = xgb_labels, max.depth = 6,eta = 1,
              nthread = 2, nrounds = 12, objective = "binary:logistic", verbose = 2)
```

```
pred6<-predict(bst6, as.matrix(data_test))
```

```
xgb_prediction6<-as.numeric(pred6>0.5)
xgb_prediction6<-as.factor(xgb_prediction6)
```

```
xgb_tree6<-confusionMatrix(xgb_prediction6,labels_test, positive ="1")
xgb_tree6
```

Árbol de Clasificación

El siguiente modelo es mediante Árboles de Clasificación donde se estudiará la activación o no activación de la opción boosting.

Cuando se habla de *boosting* se refiere a la creación de numerosos árboles de decisión y estos votan en la mejor clase para cada muestra. En este caso vamos a aplicar un boost de 10.

No boosting

```
tree_model<-C5.0(data_train,labels_train)
tree_model
```

```
summary(tree_model)
```

Creamos un vector de valores de *class* predichos, los cuales compararemos con los valores reales de clase de los datos test.

```
tree_predict<-predict(tree_model,data_test)
cfm_tree<-confusionMatrix(tree_predict,labels_test, positive ="1")
cfm_tree
```

Boosting on 15

Vamos ahora a observar el rendimiento del modelo teniendo en cuenta la opción boosting (trials) que indica el número de árboles de decisión separados para usar en la mejora del modelo. El valor de trials es 15 que es el que se suele utilizar por defecto

```
set.seed(12345)
tree_model_boost<-C5.0(data_train,labels_train,trials = 15)
```

Como podemos observar se ha producido una gran mejora en el rendimiento del modelo obteniendo ahora solo 26 fallos con una tasa de error de 9.9%.

```
set.seed(12345)
tree_boost_predict<-predict(tree_model_boost,data_test)
cfm_tree_boost<-confusionMatrix(tree_boost_predict,labels_test, positive ="1")
cfm_tree_boost
```

Boosting on 40

```
set.seed(12345)
tree_model_boost40<-C5.0(data_train,labels_train,trials = 40)
```

```
set.seed(12345)
tree_boost_predict40<-predict(tree_model_boost40,data_test)
cfm_tree_boost40<-confusionMatrix(tree_boost_predict40,labels_test, positive ="1")
cfm_tree_boost40
```

```
tree_boost_predict40<-predict(tree_model_boost40, data_test, type="prob")
ROC_tree<-roc(labels_test, tree_boost_predict40[,2])
ROC_tree_auc<-auc(ROC_tree)
ROC_tree_auc
```

Boosting on 50

```
set.seed(12345)
tree_model_boost50<-C5.0(data_train,labels_train,trials = 50)
```

```
set.seed(12345)
tree_boost_predict50<-predict(tree_model_boost50,data_test)
cfm_tree_boost50<-confusionMatrix(tree_boost_predict50,labels_test, positive ="1")
cfm_tree_boost50
```

Random Forest

Se usan de ejemplo número de árboles de 50, 100 y 200

```
set.seed(12345)
training<-sample(272,floor(272*0.670))
rf_train<-Data_[training,]
rf_test<-Data_[-training,]
```

Random Forest = 50

Random Forest (50)

```
set.seed(12345)
rf_model_50<-randomForest(Activity~., data = rf_train, ntree=50)
print(rf_model_50)
```

```
predict_rf_50<-predict(rf_model_50, rf_test[-2])
confusionMatrix(rf_test$Activity, predict_rf_50, positive = "1")
```

Random Forest = 100

Random Forest (100)

```
set.seed(12345)
rf_model_100<-randomForest(Activity~., data = rf_train, ntree=100)
print(rf_model_100)
```

```
plot(rf_model_100)
```

```
varImpPlot(rf_model_100, sort=TRUE, n.var = 12,nrow(rf_model_100$importanceSD),
           main = "Importancia descriptores")
```

Ver posibles distribuciones

```
par(mfrow=c(2,2))
boxplot(Data_$SlogP~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="SlogP")
boxplot(Data_$smr_VSA3~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="smr_VSA3")
boxplot(Data_$Energy~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="Energy")
boxplot(Data_$peoe_VSA7~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="peoe_VSA7")
boxplot(Data_$peoe_VSA8~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="peoe_VSA8")
boxplot(Data_$MQN29~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="MQN29")
boxplot(Data_$NumHeavyAtoms~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="NumHeavyAtoms")
boxplot(Data_$slogp_VSA6~Data_$Activity, col="red",
        xlab="Activity", ylab="Valor", main="slogp_VSA6")
```

Para la predicción del modelo:

```
predict_rf_100<-predict(rf_model_100, rf_test[-2])
confusionMatrix(rf_test$Activity, predict_rf_100, positive = "1")
```

```
predict_rf_100<-predict(rf_model_100, rf_test, type="prob")
```

```
ROC_rf<-roc(rf_test$Activity, predict_rf_100[,2])
ROC_rf_auc<-auc(ROC_rf)
ROC_rf_auc
```

```
par(pty="s")
plot(ROC_rf, col="green", legacy.axes=TRUE,percent=TRUE)
```

Random Forest = 150

```
set.seed(12345)
rf_model_150<-randomForest(Activity~., data = rf_train, ntree=150)
print(rf_model_150)
```

```
predict_rf_150<-predict(rf_model_150, rf_test[-2])
confusionMatrix(rf_test$Activity, predict_rf_150, positive = "1")
```

Random Forest = 200

Random Forest (200)

```
set.seed(12345)
rf_model_200<-randomForest(Activity~., data = rf_train, ntree=200)
print(rf_model_200)
```

```
predict_rf_200<-predict(rf_model_200, rf_test[-2])
confusionMatrix(rf_test$Activity, predict_rf_200, positive = "1")
```

Curva ROC

```
par(pty="s")
plot.roc(ROC_rf, col="green", legacy.axes=TRUE,percent=TRUE,
         xlab="Porcentaje de falsos positivos",
         ylab="Porcentaje de verdaderos positivos",lwd=2, print.auc = TRUE,
         print.auc.y = 0.1, print.auc.cex = 0.75)
plot.roc(ROC_tree, col="blue", lwd=2, add=TRUE, print.auc = TRUE,
         print.auc.y = 0.6, print.auc.cex = 0.75 )
plot.roc(ROC_xgb, col="red",lwd=2, add=TRUE, print.auc = TRUE,
         print.auc.y = 0.5, print.auc.cex = 0.75 )
plot.roc(ROC_svm, col="purple", lwd=2,add=TRUE, print.auc = TRUE,
         print.auc.y = 0.4, print.auc.cex = 0.75 )
plot.roc(ROC_nb, col="brown", lwd=2, add=TRUE, print.auc = TRUE,
         print.auc.y = 0.3, print.auc.cex = 0.75)
plot.roc(ROC_knn, col="orange", lwd=2, add=TRUE,print.auc = TRUE,
         print.auc.y = 0.2, print.auc.cex = 0.75 )
legend("bottomright",legend=c("RF", "DT", "XGB", "SVM", "NV", "k-NN"),
col=c("green","blue","red","purple","brown","orange"),
lwd=2, cex=0.5)
```

Optimización de hiperparámetros

```
customRF <- list(type = "Classification", library = "randomForest", loop = NULL)
customRF$parameters <- data.frame(parameter = c("mtry", "ntree"),
                                  class = rep("numeric", 2), label = c("mtry", "ntree"))
customRF$grid <- function(x, y, len = NULL, search = "grid") {}
customRF$fit <- function(x, y, wts, param, lev, last, weights,
                         classProbs, ...) {
  randomForest(x, y, mtry = param$mtry, ntree=param$ntree, ...)
}
customRF$predict <- function(modelFit, newdata, preProc = NULL,
                             submodels = NULL)
   predict(modelFit, newdata)
customRF$prob <- function(modelFit, newdata, preProc = NULL,
                          submodels = NULL)
   predict(modelFit, newdata, type = "prob")
customRF$sort <- function(x) x[order(x[,1]),]
customRF$levels <- function(x) x$classes
```

```
# train model
control <- trainControl(method="repeatedcv", number=10, repeats=3)
tunegrid <- expand.grid(.mtry=c(5:25), .ntree=c(50,75,100,125,150,175,200,250))
set.seed(12345)
custom <- train(Activity~., data=rf_train, method=customRF, metric="AUC",
                tuneGrid=tunegrid, trControl=control)
summary(custom)
plot(custom)
```

```
print(custom)
```

train model

```
set.seed(12345)
rf_model_op<-randomForest(Activity~., data = rf_train, ntree=75, mtry=22)
print(rf_model_op)
```

```
predict_rf_op<-predict(rf_model_op, rf_test[-2])
confusionMatrix(rf_test$Activity, predict_rf_op, positive = "1")
```

```
predict_rf_op<-predict(rf_model_op, rf_test, type="prob")
```

```
ROC_rf<-roc(rf_test$Activity, predict_rf_op[,2])
ROC_rf_auc<-auc(ROC_rf)
ROC_rf_auc
```