

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencia y Tecnología).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo o titular de los Derechos Autor.

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



**Doctorado en Ciencias
en Ciencias de la Computación**

**Búsqueda de un conjunto óptimo de descriptores moleculares para la
modelación QSAR**

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

Presenta:

Luis Antonio García González

Ensenada, Baja California, México
2023

Tesis defendida por
Luis Antonio García González

y aprobada por el siguiente Comité

Dr. Carlos Alberto Brizuela Rodríguez
Codirector de tesis

Dr. César Raúl García Jacas
Codirector de tesis

Dr. Hugo Homero Hidalgo Silva

Dr. Ansel Yoan Rodríguez González

Dr. Sergio Andrés Águila Puentes

Dr. José Luis Medina Franco



Dr. Pedro Gilberto López Mariscal
Coordinador del Posgrado en Ciencias de la Computación

Dra. Ana Denise Re Araujo
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Luis Antonio García González** como requisito parcial para la obtención del grado de Doctor en Ciencias en de la Computación.

Búsqueda de un subconjunto óptimo de descriptores moleculares para la modelación QSAR

Resumen aprobado por:

Dr. Carlos Alberto Brizuela Rodríguez

Codirector de tesis

Dr. César Raúl García Jacas

Codirector de tesis

En la actualidad, se estima que más de 10 millones de vertebrados son utilizados cada año en estudios toxicológicos. Dadas estas circunstancias, varias agencias regulatorias están impulsando activamente a la comunidad científica para el desarrollo de una alternativa a la experimentación con animales. Entre las alternativas existentes se pueden encontrar los estudios in-silico, especialmente los métodos de Relación Cuantitativa Estructura-Actividad (QSAR por sus siglas en inglés), los cuales se destacan como uno de los más utilizados. Los estudios QSAR se basan en la hipótesis de que compuestos estructuralmente similares presentan una actividad similar, lo que permite predecir la actividad de nuevos compuestos en función de compuestos estructuralmente similares, para los cuales se definió su actividad de forma experimental. Estudios han demostrado que la selección del subconjunto “óptimo” de las variables (descriptores moleculares) que caracterizan estructuralmente los compuestos tiene mayor importancia para la construcción de un modelo QSAR robusto que la estrategia de modelación utilizada. Actualmente, los descriptores moleculares (DMs) utilizados para la modelación QSAR son calculados con herramientas computacionales que no tienen en cuenta si estos caracterizan bien la actividad que se quiere modelar y los compuestos que se están analizando. En este trabajo se describen las limitaciones del enfoque actual, teniendo en cuenta que, si se sigue este enfoque, se puede pasar por alto información relevante al suponer que el conjunto de DMs calculado caracteriza bien las estructuras químicas que se están analizando, cuando en realidad puede que esto no suceda. Estas limitaciones se deben principalmente a que dichas herramientas limitan el número de DMs que calculan, restringiendo el dominio de los parámetros en los que se definen los algoritmos que calculan los DMs, parámetros que definen el Espacio de Configuración de Descriptores (DCS por sus siglas en inglés). En este trabajo se propone relajar estas restricciones en un enfoque DCS abierto, de manera que se pueda considerar inicialmente un universo más amplio de DMs y que estos caractericen de manera adecuada las estructuras a modelar. La generación de DMs se aborda entonces como un problema de optimización multicriterio, y para darle solución, dos algoritmos evolutivos son propuestos. Estos algoritmos incluyen conceptos de coevolución cooperativa para medir la sinergia entre descriptores moleculares teóricamente diferentes. Además, como componente novedoso, se propone determinar la aptitud individual de cada DM mediante la agregación de cuatro criterios utilizando la Integral de Choquet junto a una medida difusa no aditiva. Los resultados experimentales obtenidos en conjuntos de datos químicos de referencia muestran que los modelos construidos a partir de un conjunto de descriptores moleculares “optimizado” presentan una mayor capacidad predictiva que los modelos construidos a partir de un subconjunto de DM sin optimizar. En conclusión, los algoritmos propuestos resultan más adecuados para la modelación QSAR que el enfoque actualmente aplicado para obtener conjuntos de Descriptores Moleculares (MDs).

Palabras clave: algoritmos genéticos, descriptores moleculares QuBiLS-MAS, QSAR, DILI, cooperación coevolutiva

Abstract of the thesis presented by **Luis Antonio García González** as a partial requirement to obtain the Doctor of Science degree in Computer.

Search for an optimal subset of molecular descriptors for QSAR modeling

Abstract approved by:

Ph.D. Carlos Alberto Brizuela Rodríguez
Thesis Director

Ph.D. César Raúl García Jacas
Thesis Director

Currently, it is estimated that more than 10 million vertebrates are used per year for toxicological studies. Numerous regulatory agencies are actively advocating for the development of alternative methods to avoid unnecessary experimentation on animals. Among the existing alternatives in silico studies, especially Quantitative Structure Activity Relationships (QSAR) methods, stands out as one of the most widely used approaches. QSAR Methods are based on the premise that molecules with similar structures presents similar activities, which makes it possible to predict the activity of new compounds based on structurally similar compounds, for which their activity has been defined experimentally. Studies have demonstrated that the selection of the “optimal” set of molecular descriptors (MDs) is more important to build a robust QSAR models than the choice of the learning algorithm. Nowadays, the molecular descriptors (MD) used for QSAR modeling are calculated using computational tools that do not consider whether they accurately characterize the activity to be modeled and the compounds being analyzed. We demonstrate here that this approach may miss relevant information by assuming that the initial universe of MDs codifies, when it does not, all relevant aspects for the respective learning task. We argue that the limitation is mainly because of the constrained intervals of the parameters used in the algorithms that compute the MDs, parameters that define the Descriptor Configuration Space (DCS). We propose to relax these constraints in an open CDS approach, so that a larger universe of MDs can initially be considered, and these descriptors can adequately characterize the structures to be modeled. We model the MD generation as a multicriteria optimization problem, and two genetic algorithms-based approaches are proposed to solve it. These algorithms include cooperative-coevolutionary concepts to consider the synergism between theoretically different MDs during the evolutionary process. As a novel component, the individual fitness function is computed by aggregating four criteria via the Choquet Integral using a fuzzy non-additive measure. Experimental outcomes on benchmarking chemical datasets show that models created from an “optimized” sets of MDs present greater probability to achieve better performances than models created from sets of MDs obtained without optimizing their DCSs. Therefore, it can be concluded that the proposed algorithms are more suitable than the approach currently applied to obtain sets of MDs for QSAR modeling.

Keywords: genetics algorithms, QuBiLS-MAS molecular descriptors, QSAR, DILI, cooperative-coevolutionary algorithms

Dedicatoria

Para Adry, por su lucha contra lo injusto.

Agradecimientos

Al Consejo Nacional de Humanidades Ciencias y Tecnologías por el apoyo económico para realizar mis estudios.

Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California y al Departamento de Ciencias de la Computación del CICESE, por el apoyo.

A mis compañeros del CEMC por ser el mejor centro de investigación.

A mis asesores por su empeño en hacerme mejor investigador.

A los profesores Hugo Hidalgo Silva, Ansel Yoan Rodríguez González, Sergio Águila Puentes y José Luis Medina Franco por las recomendaciones oportunas que contribuyeron a mejorar mi investigación.

A Tere y Arnulfo por abrirme la puerta al México lindo y querido.

A Loge y Mónica por el cariño brindado.

A Yulith, Bere y Mitch por mostrarme que un fin de semana puede comenzar un martes.

A Occidental #1 y a la masonería bajacaliforniana por mostrarme lo que quiero y no quiero ser.

A Yovani Marrero Ponce, por mostrarme que un genio puede ser modesto.

A mi familia por malcriarme.

A mi hermana por ser la mejor hija del mundo.

A mis padres por el amor incondicional.

A César y Lisset por alejar la soledad.

A Adriana por mantener la sonrisa.

En fin, a los muertos de mi felicidad.

Tabla de contenido

	Página
Resumen en español.....	ii
Resumen en inglés.....	iii
Dedicatoria	iv
Agradecimientos.....	v
Lista de figuras.....	ix
Lista de tablas	x
Lista de Algoritmos	xii
Capítulo 1. Introducción.....	1
1.1 Antecedentes	3
1.2 Hipótesis.....	6
1.3 Objetivos	6
1.3.1 Objetivo general	7
1.3.2 Objetivos específicos.....	7
1.4 Contribuciones	7
1.5 Estructura de la tesis	8
Capítulo 2. Marco teórico.....	9
2.1 Descriptores moleculares.....	9
2.1.1 Descriptores QuBiLS-MAS	11
2.2 Modelación QSAR.....	14
2.3 Algoritmos genéticos.....	17
2.3.1 Algoritmos coevolutivos.....	18
2.4 Medidas e integrales difusas.....	19

2.4.1	Medidas difusas.....	20
2.4.2	Integrales difusas	21
2.5	Conclusiones parciales	22
Capítulo 3. Algoritmo para Explorar y Optimizar el Espacio de Configuración del Descriptor 23		
3.1	Función para determinar la función de calidad de los descriptores	23
3.1.1	Métricas para determinar la calidad de los descriptores moleculares.....	23
3.1.2	Operador de agregación basada en la Integral de Choquet para determinar la calidad global de los descriptores.....	26
3.2	Función para determinar la calidad de los subconjuntos	27
3.3	AExOp-DCS: Algoritmo para Explorar y Optimizar el Espacio de Configuración del Descriptor ...	28
3.4	Validación del algoritmo AExOp-DCS en la construcción de modelos QSAR a partir de los subconjuntos retornados.	33
3.4.1	Bases para la predicción de actividad biológica.....	33
3.4.2	Procedimiento para validar la calidad de los DMs retornados	35
3.4.3	Rendimiento de los modelos obtenidos	36
3.4.4	Estudio estadístico basado en estimación bayesiana	41
3.4.5	Análisis de los mejores modelos por cada conjunto de datos.....	43
3.5	Conclusiones parciales	45
Capítulo 4. Un enfoque coevolutivo para determinar un conjunto óptimo de descriptores moleculares mediante la exploración y optimización del Espacio de Configuración del Descriptor46		
4.1	C-AExOp-DCS: Algoritmo Coevolutivo para Explorar y Optimizar el Espacio de configuración del Descriptor	46
4.2	Validación del algoritmo C-AExOp-DCS en la construcción de modelos QSAR.	49
4.2.1	Procedimiento para validar la calidad de los descriptores moleculares retornados.....	49
4.2.2	Evaluación de los modelos construidos y análisis respecto a modelos definidos en la literatura.....	50

4.2.3	Estudio estadístico basado en estimación bayesiana	54
4.2.4	Análisis de los mejores modelos por cada conjunto de datos	56
4.3	Conclusiones parciales	57
Capítulo 5. Desarrollo de modelos para la predicción de actividad hepatotóxica.....		59
5.1	Bases para la predicción de actividad hepática	59
5.2	Procedimientos para construir y validar los modelos para la predicción de DILI	60
5.3	Rendimiento de los modelos construidos en el conjunto de entrenamiento	62
5.4	Capacidad de generalización de los modelos construidos.....	64
5.4.1	Análisis del rendimiento de los modelos en el conjunto de prueba Liew_3_ValPair_20 .	68
5.5	Análisis de la diversidad y complejidad de los conjuntos analizados	69
5.6	Conclusiones parciales	72
Capítulo 6. AExOp-DCS: Software para Explorar y Optimizar el Espacio de Configuración del Descriptor		73
6.1	Dependencias y ambiente de trabajo	74
6.2	Descripción de la Interfaz de usuario por la línea de comandos (<i>CLI</i>)	74
6.3	Descripción de la Interfaz Gráfica de Usuario (<i>GUI</i>).....	75
6.4	Interfaz de Programación Abstracta (<i>API</i>).....	77
6.5	Conclusiones parciales	79
Capítulo 7. Conclusiones y trabajo futuro		81
7.1	Conclusiones.....	81
7.2	Trabajo futuro	81
Literatura citada		83
Anexos		99

Lista de figuras

Figura	Página
1. Distribución de la SE para los DMs RDF calculados por PaDEL (Yap, 2011) utilizando un paso igual a 0.5 (valor por omisión) y 0.1 (valor modificado) para obtener todos los posibles valores de R . El valor de la entropía fue calculado luego de evaluar los DMs sobre el conjunto de entrenamiento de Liew (Liew et al., 2011) de compuestos con actividad relacionada a la Toxicidad Inducida por Medicamentos (DILI por sus siglas en inglés).....	5
2. Flujo básico para el cálculo de los descriptores QuBiLS-MAS (Valdés-Martín et al., 2017)....	12
3. Flujo básico de la modelación QSAR	15
4. Definición del cromosoma para representar los DMs RDF. (A) Representación del cromosoma, (B) Ejemplo de una población con cromosomas que representan DMs RDF, (C) Evaluaciones de los cromosomas dentro de la población en un conjunto de moléculas de ejemplo.	29
5. Definición del cromosoma para representar los DMs 3D-MoRSE. (A) Representación del cromosoma, (B) Ejemplo de una población con cromosomas que representan DMs 3D-MoRSE, (C) Evaluaciones de los cromosomas dentro de la población en un conjunto de moléculas de ejemplo.	30
6. Ejemplo de (A) cálculo de la calidad del DM sin considerar DMs de otras poblaciones. (B) Cálculo de la calidad de un DM considerando DMs de otras poblaciones.....	49
7. Entropía de Shannon normalizada de las frecuencias de los <i>scaffold</i> por cada conjunto de datos.....	70
8. Puntaje espacial por cada conjunto de datos.	71
9. Diagrama básico del software AExOp-DCS.....	73
10. Ventana principal del software AExOp-DCS.....	76
11. Diagrama UML para las clases en las que están implementada la lógica principal	78
12. Diagrama UML para las clases del paquete <i>configurations</i> , donde se define la configuración general para la ejecución del algoritmo.....	79
13. Estructura química de los compuestos incluidos en el conjunto de datos Liew_3_ValPair_20	103
14. Estructura química de los compuestos incluidos en el conjunto de datos Liew_3_ValPair_20	104

Lista de tablas

Tabla	Página
1. Descripción de los ocho conjuntos de compuestos químicos propuestos por Sutherland.	34
2. Descripción de los seis conjuntos de compuestos químicos relacionados con actividad ecotoxicológica.....	34
3. Configuración utilizada por ambos algoritmos para determinar el subconjunto óptimo de DMs.	36
4. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos construidos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de Sutherland.....	37
5. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los mejores modelos construidos respecto a los modelos reportados en la literatura en el conjunto de datos de ecotoxicidad.....	38
6. Habilidades de generalización Qtest2 obtenidos por los mejores modelos construidos respecto los mejores modelos reportados en la literatura para el conjunto de datos de Sutherland.....	40
7. Habilidades de generalización Qtest2 obtenidos por los mejores modelos construidos y los mejores modelos reportados en la literatura para los conjuntos de compuestos ecotoxicológicos.....	41
8. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos no cooperativos respecto a los mejores modelos reportados en la literatura.....	42
9. Valores de rendimiento en entrenamiento, validación y DA alcanzados por los mejores modelos en cada conjunto de datos.....	44
10. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos cooperativos y no cooperativos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de Sutherland.	51
11. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos cooperativos y no cooperativos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de ecotoxicidad.	52
12. Habilidades de generalización Qtest2 obtenidos por los mejores modelos no cooperativos y cooperativos y los mejores modelos reportados en la literatura para el conjunto de datos de Sutherland.....	53
13. Habilidades de generalización Qtest2 obtenidos por los mejores modelos no cooperativos y cooperativos y los mejores modelos reportados en la literatura para los conjuntos de compuestos ecotoxicológicos.....	54

14. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos cooperativos respecto a los mejores modelos reportados en la literatura. 55
15. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos no cooperativos respecto a los modelos cooperativos..... 55
16. Valores de rendimiento en entrenamiento, validación y DA alcanzados por los mejores modelos cooperativos en cada conjunto de datos. 58
17. Conjuntos de datos utilizados para la construcción de modelos para la predicción de DILI... 60
18. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Liew_0_TR_1075 63
19. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Nguyen_0_TR_1596..... 63
20. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Schyman_0_TR_1423..... 64
21. Comparación del rendimiento según MCC en siete conjuntos de prueba de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI..... 65
22. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de prueba Liew_1_R_TS_120 y Liew_2_B_TS_47, así como en sus respectivos DA..... 99
23. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de prueba Mora_4_ETS_554 y Nguyen_1_TS_322, así como en sus respectivos DA 100
24. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de Nguyen_2_TS_52 y Garcia_1_TS_106, así como en sus respectivos DA 101
25. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en el conjunto Liew_3_ValPair_20, así como en su respectivo DA..... 102

Lista de Algoritmos

Algoritmo	Página
1. Algoritmo genético.....	17
2. Algoritmo genético coevolutivo cooperativo.....	19
3. Algoritmo para determinar la calidad según <i>ReliefF</i> para cada uno de los DMs.....	25
4. Seudo código del algoritmo AExOp-DCS.	32
5. Seudo Código del algoritmo C-AExOp-DCS.	47

Capítulo 1. Introducción

El descubrimiento de fármacos asistido por computadora (CADD por sus siglas en inglés) constituye un prominente enfoque de investigación dirigido al diseño, optimización y selección de compuestos biológicamente activos mediante el uso de herramientas computacionales (P. Lu et al., 2018; Muegge et al., 2017; Oglic et al., 2018; Sass, 2017). Este proceso integra varias disciplinas científicas, cuyas actividades contribuyen a minimizar los costos de investigación-desarrollo y el lapso entre la idea (i.e. identificación del objetivo) y la obtención del fármaco para dar “respuesta” a diferentes patologías. Estos aspectos constituyen retos actuales de la industria farmacéutica y biotecnológica con el fin de lograr mayor productividad (Oglic et al., 2018; Sass, 2017) en un entorno donde existen altas tasas de mortalidad debido a enfermedades como las cardiovasculares y neoplásicas, así como infecciones de naturaleza viral y parasitaria.

Un estudio reciente muestra como entre los años 2010 a 2019 solo fueron aprobados un total de 268 nuevos fármacos (Brown & Wobst, 2021). En promedio el tiempo transcurrido entre la identificación del candidato a fármaco hasta la aprobación por la agencia regulatoria fue de 8.7 (± 3.8) años, existiendo fármacos con más de 16 años de desarrollo (Brown & Wobst, 2021) y con un costo en promedio por encima de los miles de millones de dólares (Benfenati, 2022). Todo esto, sin contar que el 90% de los candidatos a fármacos nunca llegaron a liberarse (Benfenati, 2022).

Por otro lado, cada vez son más las organizaciones y agencias regulatorias que se posicionan en contra del uso de animales para realizar pruebas toxicológicas (Benfenati, 2022; Grimm, 2019). Se estima que más de 10 millones de vertebrados son utilizados anualmente para realizar estos experimentos (Benfenati, 2022). Por tal motivo, la Agencia de Protección Ambiental (EPA por sus siglas en inglés) anunció que dejará de apoyar a proyectos que incluyan experimentación en mamíferos a partir del año 2035, y que reducirán el apoyo a aquellos proyectos que incluyan experimentaciones en las que se utilicen aves o peces (Grimm, 2019). Además, destinó más de 4 millones de dólares para el financiamiento de estudios que no necesiten el uso de animales (Grimm, 2019). Esto implica que las técnicas computacionales, también conocidas como “estudios in-silico”, se hagan cada vez más necesarias y jueguen un papel predominante dentro del descubrimiento y desarrollo de fármacos.

Entre las técnicas dentro de los estudios in-silico se encuentran los estudios de relación cuantitativa estructura-actividad (QSAR por sus siglas en inglés). Estos se basan en la hipótesis de que las estructuras

moleculares presentan características responsables de sus propiedades fisicoquímicas y biológicas, y que estas características debieran ser codificadas en uno o más descriptores moleculares, por lo que compuestos similares debieran presentar actividad similar. Un modelo QSAR es el resultado final de un proceso que comienza con una caracterización adecuada de un conjunto de moléculas y termina con alguna inferencia, hipótesis o predicción acerca de la actividad biológica o propiedades químico-física de nuevos compuestos (Gasteiger, 2003). Utilizando los modelos QSAR, la actividad biológica (o propiedad) de un nuevo compuesto puede inferirse a partir de compuestos estructuralmente similares cuyas actividades se conocen experimentalmente (Mauri et al., 2016; Neves et al., 2018; Yousefinejad & Hemmateenejad, 2015).

En general, para el desarrollo de un modelo QSAR se requiere de un conjunto de moléculas con actividad biológica conocida (la variable dependiente del modelo), de un conjunto de descriptores moleculares (variables independientes del modelo) que codifiquen información relevante de las moléculas bajo estudio, y de la aplicación de un método matemático/computacional para definir una relación entre la actividad y la estructura de los compuestos (Mauri et al., 2016; Timbrell, 2008; D. S. Wishart, 2007; Yousefinejad & Hemmateenejad, 2015). La modelación QSAR se ha aplicado satisfactoriamente para la predicción de diferentes actividades/propiedades, tales como estudios antibacteriales (Dolezal et al., 2016; Faidallah et al., 2018; Gholivand et al., 2016; Hodyna et al., 2016; Leemans et al., 2016; Panda et al., 2015), anticancerígenos (da Silva et al., 2015; Deep et al., 2016; Pingaew et al., 2015; Sławiński et al., 2017; Tugcu et al., 2019; J. Wang et al., 2018), antituberculosos (Akrami & Niazi, 2017; Pinto et al., 2019; P. P. Roy et al., 2018; Sahasrabudhe et al., 2015; Sharapova et al., 2017; Tseng et al., 2017), anticonvulsivos (Antanasijević et al., 2017; Bellera & Talevi, 2019; Chen et al., 2016; Garro Martinez et al., 2015; Pradhan & Goyal, 2016), antimicrobiales (Abdelrahman et al., 2017; Al-Fakih et al., 2019; Bai et al., 2019; Božić et al., 2018; Deep et al., 2016; Ghanem et al., 2018; Kawczak et al., 2018; Lino et al., 2018; Pizzolitto et al., 2020; A. P. Toropova et al., 2017; M. A. Toropova et al., 2015; Trush et al., 2019), entre otros (J. Lu et al., 2014; Xu et al., 2017).

Uno de los problemas más estudiados en la modelación QSAR son los relacionados con la predicción de actividad toxicológica (Ballabio et al., 2019; Dhingra et al., 2019; El-Zahabi et al., 2019; Gökçe & Saçan, 2019; Kaitoh et al., 2019; Klüver et al., 2019; Kuz'min et al., 2019; Nagai et al., 2019; P. P. Roy et al., 2019; Sheffield & Judson, 2019; A. P. Toropova et al., 2017; D. Wang et al., 2018). Esto se debe a que una de las principales causas de que un candidato a fármaco no sea liberado es la presencia de esta actividad (Benfenati, 2022).

1.1 Antecedentes

Un paso fundamental en la construcción de los modelos QSAR es la selección del mejor conjunto de descriptores moleculares (DMs) (Mauri et al., 2016). Esto se debe a que no existe conocimiento a priori sobre cuáles son los mejores DMs para desarrollar un modelo QSAR específico. Un descriptor molecular es el resultado final de un procedimiento matemático que transforma información codificada dentro de una representación simbólica de una molécula en un número de utilidad (Mauri et al., 2016). Los descriptores moleculares están basados en diferentes teorías y enfoques, tales como química-cuántica, teoría de información, y teoría de grafos (Gasteiger, 2003; Mauri et al., 2016; K. Roy, 2020; Todeschini et al., 2009, 2020). Evidencia de la relevancia de los descriptores moleculares es la cantidad de descriptores propuestos en los últimos años (Consonni & Todeschini, 2010; Gasteiger, 2003; Grisoni et al., 2018; Mauri et al., 2016), así como las diferentes herramientas computacionales desarrolladas para su cálculo (García-Jacas et al., 2014; Mauri et al., 2006; Valdés-Martini et al., 2017; Yap, 2011).

En la actualidad, el enfoque de modelación consiste en primero calcular un número elevado de DMs, que pueden ser relevantes o no para las estructuras químicas y actividad biológica bajo estudio, para finalmente seleccionar los mejores de acuerdo con diferentes métodos de selección de rasgos (Bolón-Canedo & Alonso-Betanzos, 2019; Pes, 2020). El cálculo de muchos DMs y el uso de más de un algoritmo de selección conducen inevitablemente a un alto costo computacional, lo que implica mayor esfuerzo y tiempo. Sin embargo, calcular pocos DMs y/o considerar pocos selectores de características (o uno solo) probablemente conlleva a no determinar y utilizar los mejores DMs para un problema específico. Por lo que se hace interesante definir nuevos enfoques que obtengan los mejores DMs en dependencia del problema que se esté modelando.

Los algoritmos de cálculo de descriptores dependen de varios parámetros, tales como propiedad atómica, fragmento químico, forma algebraica, operador de agregación, entre otros. Cada uno de estos parámetros están definidos en un dominio de valores, y una combinación específica de esos valores determina un DM específico. Por ejemplo, el algoritmo para calcular los DMs Función de Distribución Radial (RDF) (Hemmer et al., 1999) recibe como parámetros una propiedad atómica (w), un factor de escala (f), un parámetro de suavizado (β), y un radio de volumen esférico (R). Por lo tanto, para $w = \text{masa atómica } (m)$, $\beta = 100\text{\AA}^{-2}$, $f = 1$ y $R = 1\text{\AA}$, se calcula un DM RDF específico denominado $RDF010m$ (notación en el software alvaDesc (Mauri, 2020)), mientras que si se mantienen los mismos valores de w , f y β pero usando un $R = 9\text{\AA}$, se calcula otro RDF DM denominado $RDF090m$. Al conjunto

de parámetros y sus valores de dominio correspondientes a un algoritmo de cálculo de descriptores se define en este trabajo como Espacio de Configuración de Descriptores (DCS, por sus siglas en inglés).

Actualmente existen diferentes herramientas tanto libres como comerciales para el cálculo de descriptores moleculares. Entre estos se encuentran alvaDesc (Mauri, 2020), Dragon (Mauri et al., 2006), PaDEL (Yap, 2011), QuBiLS-MAS (Valdés-Martín et al., 2017), QuBiLS-MIDAS (García-Jacas et al., 2014), entre otros (Moriwaki et al., 2018; Terán et al., 2019; Willighagen et al., 2017). Los algoritmos definidos actualmente para calcular DMs se implementan en softwares que pueden ser clasificados como softwares de configuración cerrada (Mauri, 2020; Mauri et al., 2006; Yap, 2011) y de configuración abierta (García-Jacas et al., 2014; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martín et al., 2017).

Por un lado, los softwares de configuración cerrada (C-DCS por sus siglas en inglés), como PaDEL (Yap, 2011) y alvaDesc (Mauri, 2020), son aquellos donde el DCS de cada algoritmo se restringe a ciertos valores determinados. Por ejemplo, en el software alvaDesc (Mauri, 2020), el DCS del algoritmo para calcular DMs RDF tiene los parámetros f y β fijados en 1 y 100\AA^{-2} , respectivamente, mientras que el parámetro w está fijado en siete ponderaciones atómicas, y el parámetro R se fija en 30 valores que van desde 1\AA hasta 15.5\AA con un paso igual a 0.5. Esto permite calcular solo 210 DMs RDF a pesar de que se pueden considerar otros valores para cada uno de los parámetros. Por ejemplo, otras ponderaciones atómicas (o propiedades) como el área de superficie polar (PSA), dureza (hardness), o suavidad (softness); otros valores de β en el intervalo entre 100\AA^{-2} y 200\AA^{-2} [13], y otros pasos (p. ej., 0.1) para calcular los valores del parámetro R . De forma similar, se pueden establecer estas consideraciones para otros algoritmos de cálculo de descriptores implementados en softwares C-DCS. Por ejemplo, 3D-MORSE (Devinyak et al., 2014), WHIM (Gramatica, 2006), entre otros (Mauri et al., 2016).

Al restringir el DCS de cada algoritmo, solo se puede obtener un subconjunto de todos los DMs posibles, por lo que es probable que se deje de codificar información química relevante en dependencia del conjunto de datos químicos que se esté analizando. Esto puede influir en la calidad de los modelos QSAR construidos. Una demostración preliminar de esta observación se puede observar en la Figura 1, donde se muestra la distribución de la Entropía de Shannon (SE) para dos subconjuntos de descriptores RDF (Barigye et al., 2013; García-Jacas et al., 2014; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Godden et al., 2000; Mora et al., 2020; Urias et al., 2015). El primer subconjunto (distribución azul en la figura) contiene los 210 DMs RDF calculados por PaDEL (Yap, 2011), los cuales se obtienen a partir de 7 propiedades atómicas (w) y 30 valores diferentes para el Radio; $R \in [1\text{\AA}, 15.5\text{\AA}]$ con un paso igual a 0.5, dejando f y β constantes. El segundo subconjunto

(distribución en rojo) es el resultado de ampliar el número de posibles valores para R , definiendo un paso de 0.1. Esto implica que R puede tomar 147 valores posibles, por lo que se pueden calcular 1029 descriptores moleculares RDF. Para no tener DMs repetidos en ambos subconjuntos, los DMs existentes en el primer subconjunto fueron eliminados del segundo.

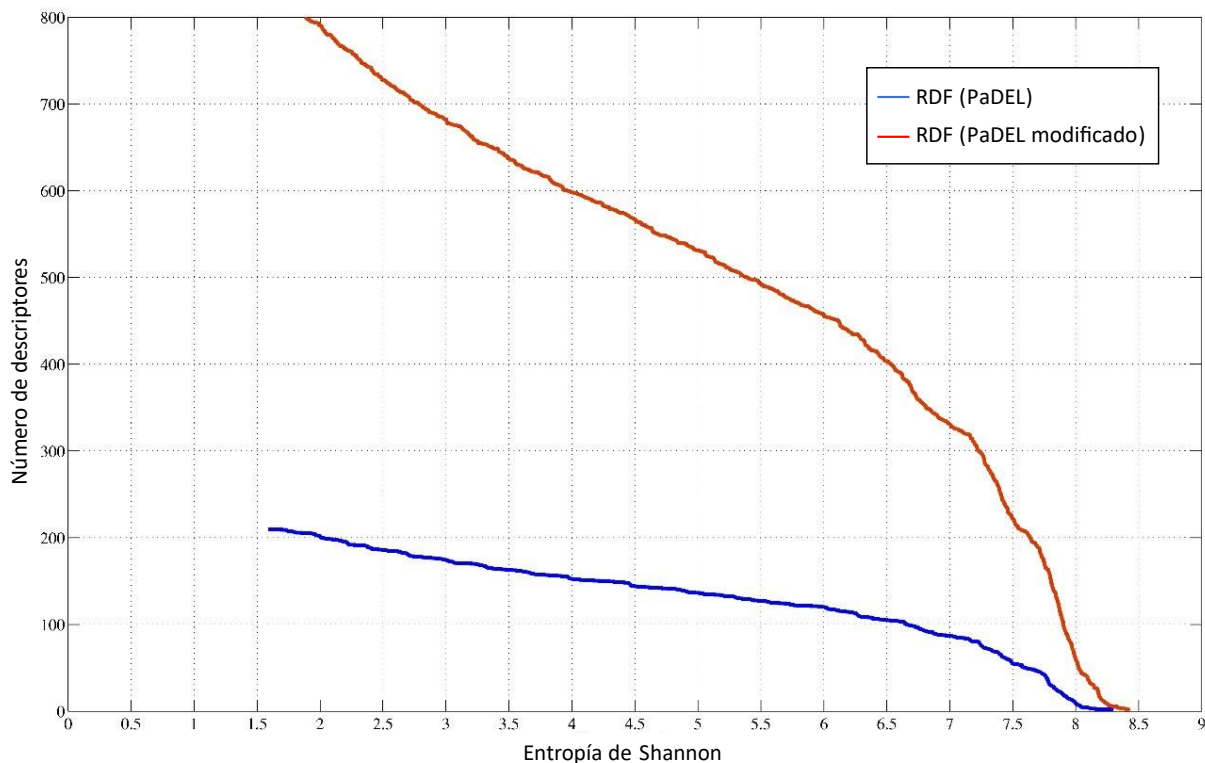


Figura 1. Distribución de la SE para los DMs RDF calculados por PaDEL (Yap, 2011) utilizando un paso igual a **0.5** (valor por omisión) y **0.1** (valor modificado) para obtener todos los posibles valores de R . El valor de la entropía fue calculado luego de evaluar los DMs sobre el conjunto de entrenamiento de Liew (Liew et al., 2011) de compuestos con actividad relacionada a la Toxicidad Inducida por Medicamentos (DILI por sus siglas en inglés).

Como un resultado, se puede notar que varios de los DMs RDF que se calcularon usando un paso igual a 0.1 (para obtener los valores del parámetro R) tienen mayor contenido de información que los DMs que se calcularon con un paso igual a 0.5 (valor fijo en el software alvaDesc (Mauri, 2020) y PaDEL (Yap, 2011)). Sin embargo, estos DMs con mayor contenido de información no se pueden considerar actualmente para crear modelos QSAR, a menos que se modifique el código fuente de los programas abiertos existentes (por ejemplo, PaDEL (Yap, 2011)) para extender el dominio de cada parámetro. Se pueden establecer conclusiones similares para otros tipos de DMs calculados con softwares C-DCS. No obstante, softwares C-DCS presentan como ventaja que el número de DMs que calculan no es grande (menos de 6 mil DMs), evitando manejar conjuntos de alta dimensionalidad.

Por otra parte, los softwares de configuración abierta (O-DCS por sus siglas en inglés), como QuBiLS-MAS(Valdés-Martín et al., 2017) y QuBiLS-MIDAS (García-Jacas et al., 2014), son aquellos en los que el DCS es personalizado por el usuario según a su experiencia (conocimiento experto). Si bien este enfoque brinda flexibilidad, tiene como principal desventaja el alto número de DMs que se pueden calcular, lo que llevaría a manejar conjuntos de datos de alta dimensionalidad. Por ejemplo, configurando el DCS de los algoritmos implementados en el software QuBiLS-MAS (Valdés-Martín et al., 2017), se pueden obtener más de 200 millones de DMs basados en átomos y enlaces. Esta cantidad aumenta cuando se consideran las opciones de cortes moleculares y quiralidad. Este volumen de DMs hace que la tarea de selección de rasgos se vuelva computacionalmente inviable. Esta alta dimensionalidad se puede evitar estableciendo un DCS reducido, pero sin garantizar la obtención de los mejores DMs para un conjunto de datos químicos específico.

La importancia de un DM depende de su capacidad para caracterizar diferentes compuestos y de qué tan bien se correlaciona con actividades (o propiedades) biológicas específicas. En consecuencia, determinar el DCS universal para cada algoritmo no es posible. Por lo tanto, tomando en consideración todo lo anteriormente planteado se tiene como problema científico el siguiente: hasta la fecha, los descriptores moleculares se obtienen desde espacios de configuración específicos, los cuales se definen sin tener en cuenta el conjunto de compuestos químicos y la actividad biológica bajo estudio y, por consiguiente, se deja de codificar información química relevante lo que limita la construcción de modelos QSAR con mejor poder predictivo.

1.2 Hipótesis

Después de evaluar el marco teórico se formuló la siguiente hipótesis de investigación: el uso de un algoritmo guiado por conjunto de datos químicos y actividad biológica en la obtención de descriptores moleculares incrementa la codificación de información relevante de las estructuras químicas y posibilita el desarrollo de modelos con mejor poder predictivo.

1.3 Objetivos

Para dar respuesta al problema científico se planteó el siguiente objetivo general y los siguientes objetivos específicos.

1.3.1 Objetivo general

Proponer un algoritmo guiado por conjunto de datos químicos y actividad biológica, mediante la optimización de espacios de configuración, que permita obtener descriptores moleculares con información química relevante con el propósito de construir modelos QSAR con mejor poder predictivo.

1.3.2 Objetivos específicos

- Definir las estrategias que determinen el mejor conjunto de descriptores moleculares acorde a las estructuras químicas y la actividad que se quiere predecir.
- Validar los descriptores moleculares en la predicción de toxicidad de compuestos orgánicos.
- Implementar una herramienta computacional multiplataforma que permita configurar y ejecutar los algoritmos propuestos en diferentes ambientes computacionales.

1.4 Contribuciones

La presente investigación permitió obtener resultados cuya novedad científica radica en: 1) la propuesta de un algoritmo basado en algoritmo genético que retorna un subconjunto “óptimo” de DMs a partir de la exploración del DCS asociado a los algoritmos de cálculo de DMs que se estén analizando; 2) la propuesta de un algoritmo, basado en coevolución cooperativa para la búsqueda de un subconjunto “óptimo” de DMs a partir de la exploración del DCS asociado a los algoritmos de cálculo de DMs que se estén analizando.

El valor metodológico de esta investigación radica en: 1) se propone modificar el flujo básico de la modelación QSAR utilizado en la actualidad, en el cual los modelos QSAR se construyen a partir de subconjuntos de DMs sin “optimizar”, mientras que nuestra investigación propone realizar la modelación QSAR utilizando conjuntos de DMs optimizados; 2) el uso de una función de calidad de DMs definida a partir de la agregación de cuatro criterios distintos mediante la integral difusa de Choquet.

Como aporte práctico se muestra la construcción de diferentes modelos para la predicción de actividad biológica a partir de los subconjuntos de DMs devueltos por los algoritmos propuestos, en particular los modelos para la predicción de lesión hepática inducida por medicamentos. Dichos modelos

van a estar disponibles de forma gratuita para la comunidad científica. Además, como resultado de la investigación se obtuvo una herramienta computacional multiplataforma en la que están implementados ambos algoritmos.

1.5 Estructura de la tesis

La presente tesis está estructurada en seis capítulos. En este primer capítulo se mencionan los antecedentes de esta investigación, así como los objetivos que guían su desarrollo. En el capítulo 2 se introducen los conceptos relacionados a la modelación QSAR, los pasos necesarios para su realización, así como las reglas necesarias para un modelado exitoso. De igual forma se define qué es un descriptor molecular y los tipos existentes de descriptor molecular a partir de la dimensión en las cuales se definen. En los capítulos 3 y 4 se definen y evalúan los algoritmos de búsqueda de descriptores moleculares propuestos en este trabajo. Por otro lado, en el capítulo 5 se proponen varios modelos para la predicción de lesión hepática inducida por medicamentos a partir de subconjuntos de DMs optimizados vía el algoritmo C-AExOp-DCS. En el capítulo 6 se describe la herramienta computacional AExOp-DCS, en la cual están implementados ambos algoritmos propuestos. Finalmente, en el capítulo 7 se presentan las conclusiones de este trabajo junto con algunas ideas de posible trabajo futuro motivado por el proyecto de investigación aquí desarrollado.

Capítulo 2. Marco teórico

En este capítulo se describen las principales características de la modelación QSAR, así como las etapas en que se divide y las reglas definidas para llevar a cabo una modelación exitosa. De cada etapa se enumeran los rasgos principales, además de mencionar las principales técnicas que se utilizan por cada una. Además, se describen las características principales de los algoritmos genéticos y del concepto de coevolución cooperativa, conceptos utilizados para el desarrollo de los algoritmos propuestos. Por otro lado, se mencionan los conceptos principales relacionados a las medidas difusas y a la Integral de Choquet, operador utilizado para determinar la calidad de los DMs. En este capítulo también se describen los descriptores QuBiLS-MAS, descriptores moleculares que serán utilizados en el desarrollo de esta investigación.

2.1 Descriptores moleculares

Un DM es el resultado final de un procedimiento matemático que transforma información codificada dentro de una representación simbólica de una molécula en un número de utilidad (Mauri et al., 2016). La habilidad de los descriptores de definir relación estructura-actividad permitió, por primera vez, relacionar conocimiento experimental con información teórica obtenida de la estructura molecular; definiendo entonces un nuevo paradigma de investigación: mientras que anteriormente, la modelación molecular consistía principalmente, en obtener la relación a partir del conocimiento experimental, actualmente la modelación QSAR se basa en propiedades determinadas computacionalmente, permitiendo el cálculo de un número mayor de características moleculares a un menor costo (Todeschini et al., 2020).

Una consideración importante a tener en cuenta en el uso de los descriptores moleculares para la modelación QSAR está relacionada al contenido de información. Esto depende del tipo de representación molecular utilizada y el algoritmo definido para calcularlo. La complejidad de los descriptores moleculares es variada, ya que abarcan desde descriptores moleculares basados en el conteo de tipos de átomos y/o fragmentos estructurales en la molécula hasta descriptores más complejos que codifican información molecular espacial (Mauri et al., 2016).

Los descriptores moleculares pueden ser clasificados por la dimensionalidad de la representación

molecular a partir de la cual son definidos y pueden ser (Mauri et al., 2016; K. Roy, 2020; Todeschini et al., 2020):

- 0D: La representación molecular más sencilla es la fórmula química, la cual brinda información básica acerca de la molécula. Esta representación es independiente de cualquier conocimiento relacionado a la conectividad de los átomos. De esta representación se puede extraer información relacionada a la característica de los átomos (por ejemplo, masa o volumen) individualmente o en su totalidad. Estos descriptores son fáciles de calcular y de interpretar, pero normalmente presentan alta degeneración: mismo valor para moléculas diferentes.
- 1D: La representación unidimensional de una molécula consiste en una lista de subestructuras moleculares (por ejemplo, tipos de átomos o grupos funcionales) y no necesita tener información acerca de la estructura completa de la molécula. Los descriptores obtenidos a partir de esta representación son útiles en estudios basados en fragmentos (Mauri et al., 2016), son fáciles de calcular y de interpretar, pero pueden presentar niveles medios de degeneración.
- 2D: La representación bidimensional de una molécula (representación topológica) define la conectividad de los átomos en la molécula en términos de la presencia y naturaleza de los enlaces químicos. Esta representación molecular contiene información útil acerca de la estructura molecular, es invariante a la translación y rotación de la molécula y evita la optimización estructural, factores que contribuyen a su amplio uso (Mauri et al., 2016).
- 3D: Un nivel adicional de complejidad puede ser agregado a los descriptores visualizándolos como un objeto tridimensional, caracterizado no solo por el tipo de átomo, conectividad y adyacencia, sino también por la configuración espacial de los átomos. La molécula entonces es definida en términos de las coordenadas espaciales (x, y, z) de los átomos. Las coordenadas espaciales pueden ser obtenidas experimentalmente (por ejemplo, cristalografía) o mediante herramientas computacionales (*ChemAxon*, s/f; *RDKit: Open-Source Cheminformatics Software*, s/f). Obtener las coordenadas 3D mediante herramientas computacionales puede acarrear problemas relacionados con la representación de la molécula por lo que el costo/beneficio del uso de los descriptores 3D debe ser evaluado detenidamente (Mauri et al., 2016).

2.1.1 Descriptores QuBiLS-MAS

Los descriptores QuBiLS-MAS son descriptores algebraicos independientes de alineamiento (García-Jacas et al., 2014; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martín et al., 2017). El acrónimo QuBiLS (del inglés Quadratic, Bilinear and N-Linear mapS) hace referencia a las formas N-lineales utilizadas para representar la relación entre 2, 3 y 4 átomos (García-Jacas et al., 2014; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martín et al., 2017). MAS (del inglés graph theoretic electronic-density Matrices and Atomic weightings) hace referencia a la matriz de densidad y el vector molecular a partir de los cuales se representa la molécula (Valdés-Martín et al., 2017). La matriz de densidad es obtenida desde un pseudografo molecular y puede ser basada en las relaciones interatómicas o entre enlaces (Valdés-Martín et al., 2017). El vector molecular es una representación de la estructura química basada en los átomos, determinando la relevancia de cada uno acorde alguna propiedad atómica (por ejemplo, masa, volumen, carga) (García-Jacas et al., 2014; Valdés-Martín et al., 2017).

La Figura 2 muestra el flujo básico para el cálculo de los descriptores QuBiLS-MAS. En el primer paso se procede a calcular los vectores moleculares ($[\bar{x}_i]^T, [\bar{y}_i]$), teniendo en cuenta la forma algebraica sobre la cual se calcularán los descriptores moleculares y el número de átomos que conforman la molécula. Los componentes de un vector molecular son valores numéricos resultantes del cálculo de una propiedad atómica en los átomos de la molécula. En dependencia de la forma algebraica los vectores moleculares pueden tomar diferentes formas. Si se tiene en cuenta una forma algebraica lineal $[\bar{x}_i]^T$ será un vector unitario como se muestra en la figura, mientras que $[\bar{y}_i]$ será calculado considerando una propiedad atómica. Si se está en presencia de una forma algebraica cuadrática, los vectores son calculados a partir de la misma propiedad atómica. Mientras que si se considera una forma algebraica bilineal los vectores moleculares son calculados a partir de propiedades atómicas distintas.

A partir del pseudografo molecular se construye la matriz de densidad (NS^1), paso número 2 de la Figura 2, la cual codifica la información topológica de la molécula. Definida la matriz de densidad de orden 1 (NS^1) es posible definir entonces la matriz de densidad de orden k (NS^k), las cuales definen las interacciones no covalentes entre los átomos de la molécula. En este punto, es importante considerar que una cualidad deseable en un DM es que sea capaz de analizar solo ciertos tipos de átomos (fragmentos químicos), ya que muchas de las propiedades o actividades biológicas dependen más de las características de estas zonas moleculares que de la molécula como un todo. Considerando esto, el paso número 3 implica determinar, si es deseado, la matriz de fragmento NS_F^k , siendo F el fragmento químico que se quiere considerar. Para esto, QuBiLS-MAS tiene en cuenta siete grupos o fragmentos químicos: aceptores de

enlaces de hidrógeno (A), átomos de carbono en cadenas alifáticas (C), donadores de enlaces de hidrógeno (D), halógenos (G), grupos metilos terminales (M), átomos de carbono en porciones aromáticas (P) y heteroátomos (X) (García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martín et al., 2017).

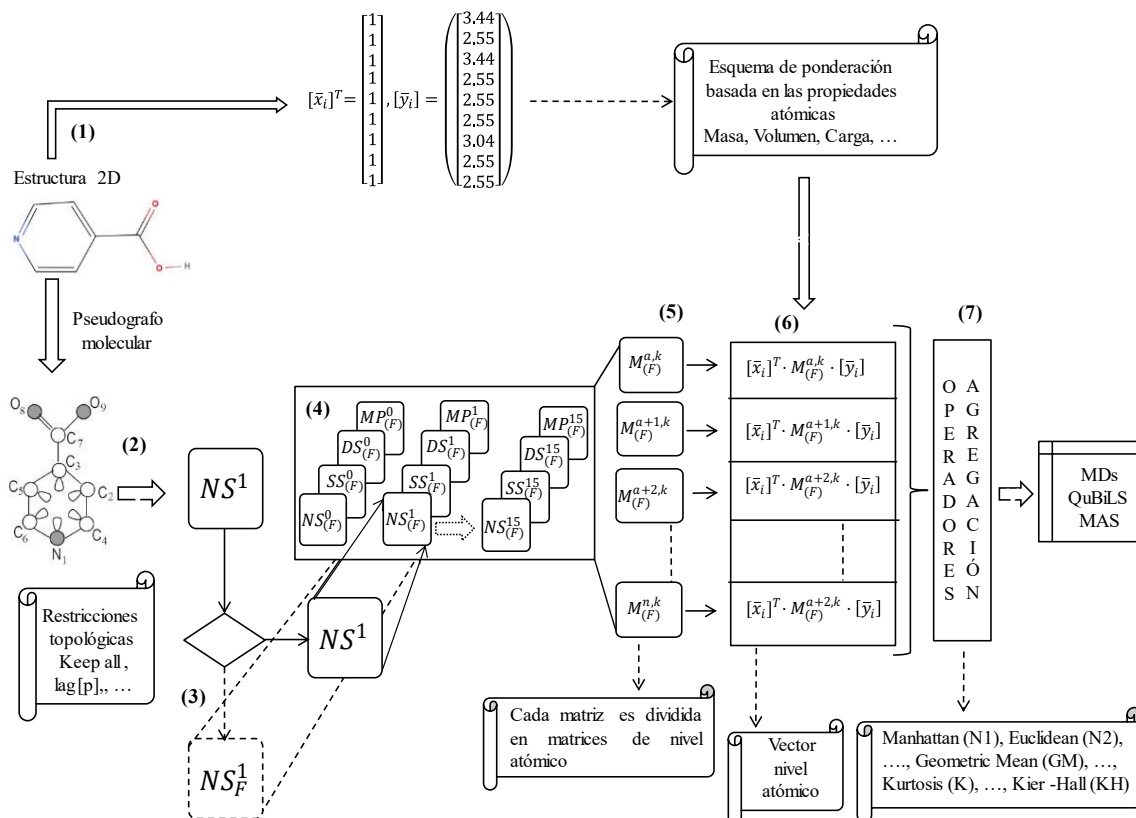


Figura 2. Flujo básico para el cálculo de los descriptores QuBiLS-MAS (Valdés-Martín et al., 2017)

Además de analizar solo ciertos grupos químicos, puede ser deseable que los DMs analicen también cierta información topológica. Para esto, los descriptores QuBiLS-MAS incluyen en la matriz NS_F^{k1} información relacionada a la distancia topológica entre sus átomos, permitiendo tener en cuenta relaciones entre átomos a corta, media y larga distancia (García-Jacas et al., 2016; Valdés-Martín et al., 2017). QuBiLS-MAS presenta tres tipos de cortes moleculares basados en la matriz de conectividad: mantener solo los elementos de la diagonal, mantener solo los elementos fuera de la diagonal y mantener solo los elementos dentro de un intervalo determinado (Valdés-Martín et al., 2017).

A las matrices definidas hasta el paso número 3, en las cuales se representa la información existente en toda la molécula (NS^k) o solamente las interacciones entre cierto grupo de átomos (NS_F^k) son

consideradas no estocásticas (ns). Estas son definidas de esta forma porque no se utiliza ningún esquema de normalización. Con el propósito de normalizar (paso número 4) las matrices definidas en los pasos anteriores, es posible ejecutar uno de los siguientes esquemas de normalización: simple estocástico (ss), doble estocástico (ds) y probabilidad mutua (pm) (García-Jacas et al., 2014; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martini et al., 2017). Las matrices resultantes de estos esquemas de normalización son denominadas SS^k , DS^k y PM^k en el caso de que se tenga en cuenta todos los átomos de la molécula, o SS_F^k , DS_F^k y PM_F^k en el caso de que se consideren solamente el fragmento químico F.

Definidas las matrices de densidad totales de orden k (SS^k , DS^k y PM^k) o las matrices de densidad de orden k para el fragmento F (SS_F^k , DS_F^k y PM_F^k), en el paso 5 se definen entonces A matrices de nivel atómico $M_F^{a,k}$, siendo A la cantidad de átomos de la molécula y a un átomo determinado. Con esta descomposición las matrices no particionadas son exactamente igual a la suma de las correspondientes matrices de nivel atómico. Definidas las matrices de nivel atómico $M_F^{a,k}$ es posible entonces definir los descriptores moleculares de nivel atómico:

$$L_a = {}_{ns[ss,ds,mp]}m_{(F)}^{a,k}(\bar{x}, \bar{y}) = [X]^T {}_{ns[ss,ds,mp]}M_{(F)}^{a,k}[Y] \quad (1)$$

$$L_e = {}_{ns[ss,ds,mp]}e_{(F)}^{a,k}(\bar{x}, \bar{y}) = [X]^T {}_{ns[ss,ds,mp]}E_{(F)}^{a,k}[Y] \quad (2)$$

donde M y E son las matrices de nivel atómico basadas en las relaciones entre átomos y enlaces covalentes respectivamente (García-Jacas et al., 2019; Valdés-Martini et al., 2017)

Una vez definido los descriptores de nivel atómico L_a o L_e , componentes del vector \bar{L} , se define el descriptor molecular a partir de un esquema de generalización mediante la combinación de las contribuciones atómicas, ver paso 7 en la

Figura 2. Este esquema de generalización obedece a un operador de agregación, no necesariamente aditivo, partiendo de la hipótesis de que la definición global más adecuada de un sistema no es necesariamente la suma de sus elementos (García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martini et al., 2017). Para esto, se utilizan diferentes operadores de agregación agrupados en 6 grupos: basados en normas de Minkoski, estadísticos de tendencia central, estadísticos de dispersión y forma, operadores clásicos, operadores GOWAWA y operadores Choquet (García-Jacas, Cabrera-Leyva, Marrero-

Ponce, Suárez-Lezcano, Cortés-Guzmán, & García-González, 2018; García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020; Valdés-Martini et al., 2017).

2.2 Modelación QSAR

La Figura 3 muestra los pasos básicos de la modelación QSAR junto a cada uno de los principios enumerados por La Organización para la Cooperación y el Desarrollo Económico (OECD por sus siglas en inglés)(OECD, s/f). El primer paso es la preparación de un conjunto de datos que contenga los datos experimentales disponibles sobre la actividad o propiedad que se quiere modelar. Este conjunto se puede conformar con compuestos ya reportados en la literatura o a partir de compuestos ya existentes en bases de datos públicas, tales como PDB (Burley et al., 2019), PubChem (S. Kim et al., 2019), DrugBank (D. S. Wishart et al., 2018), ChEMBL (Gaulton et al., 2017), Zinc (Irwin et al., 2012), entre otras (Aguilera-Mendoza et al., 2015; Banerjee et al., 2018; D. Wishart et al., 2015). Es importante señalar que la actividad de los compuestos recolectados debe haber sido determinada experimentalmente y de ser posible, utilizando el mismo proceso experimental bajo las mismas condiciones, como lo determina el Principio 1 de la OECD (OECD, s/f) .

El segundo paso implica el cálculo de un número suficientemente grande de DMs (García-Jacas et al., 2014; Mauri et al., 2006; Moriwaki et al., 2018; Terán et al., 2019; Valdés-Martini et al., 2017; Willighagen et al., 2017; Yap, 2011). Como se ha mencionado anteriormente, los DMs serán las variables independientes del problema y son los que caracterizarán los compuestos del estudio. Es necesario considerar la mayor cantidad de DMs posible, dado que no existe información a priori de cuáles son las características (DMs) que mejor describen el problema (i.e., molécula-actividad). Para calcular DMs existen varias herramientas disponibles en la literatura; tales como Dragon (Mauri et al., 2006), PaDeL (Yap, 2011) o alvaDesc (Mauri, 2020) entre otros (García-Jacas et al., 2014; *RDKit: Open-Source Cheminformatics Software*, s/f; Valdés-Martini et al., 2017). Una práctica común para validar la robustez de los modelos construidos es dividir el conjunto de compuestos en varios subconjuntos; por ejemplo, entrenamiento y prueba. El primer subconjunto es utilizado para la construcción de los modelos, mientras que el segundo se utiliza para evaluar el rendimiento de los modelos construidos (K. Roy et al., 2018; K. Roy & Ambure, 2016).

El principio 2 de la OECD hace referencia a que los modelos deben ser construidos a partir de algoritmos no ambiguos (Benfenati, 2022; OECD, s/f; K. Roy, 2020). Este principio indica la necesidad de

que tanto el proceso para la construcción del modelo, como las predicciones puedan ser reproducidas para su posterior validación y así garantizar la transparencia de las predicciones realizadas por los modelos (OECD, s/f). Para esto, diversas técnicas de modelado pueden ser aplicadas, ya sea basadas estadística clásica como métodos de aprendizaje de máquina (Ambure, Gajewicz-Skretna, et al., 2019; Ambure, Halder, et al., 2019; García-Jacas et al., 2019; García-Pereira et al., 2019; Gini et al., 2019; Rajathei et al., 2019; Shameera Ahamed et al., 2019; L. Sun et al., 2018; Terán et al., 2019; Trush et al., 2019).

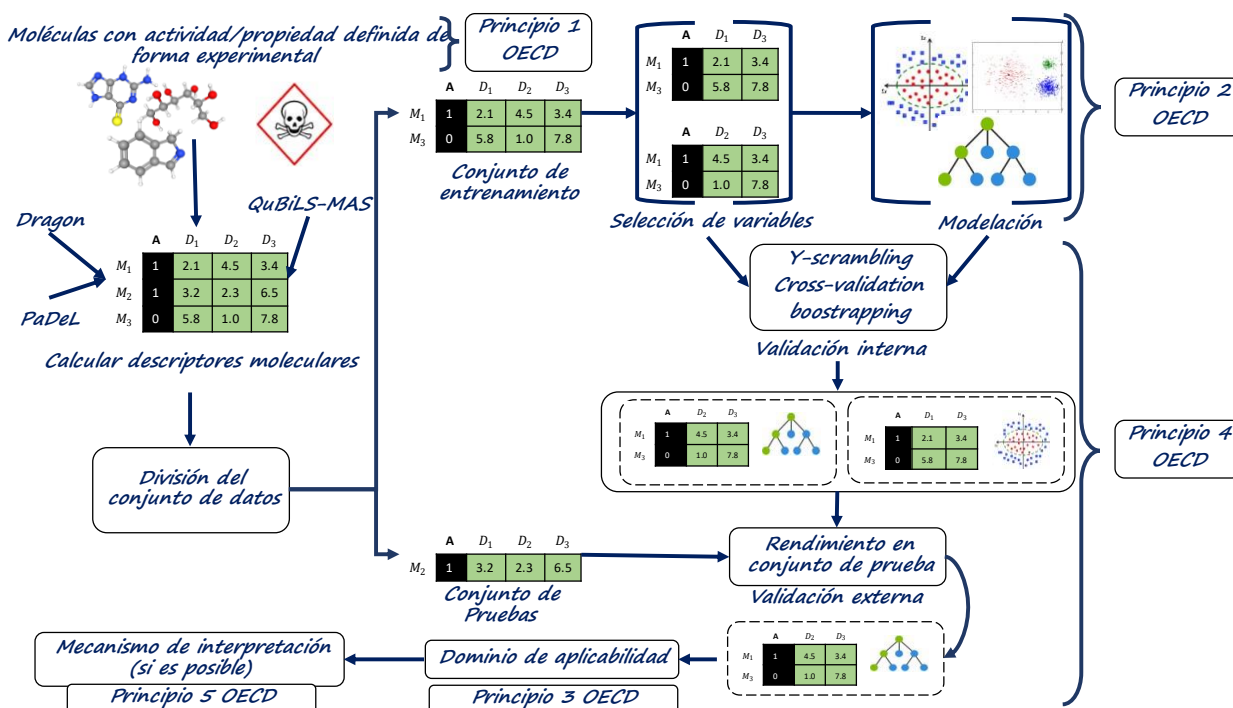


Figura 3. Flujo básico de la modelación QSAR

Es necesario recordar que antes de la construcción de los modelos es necesario realizar un paso intermedio para seleccionar los descriptores moleculares que mejor describen los compuestos dentro del conjunto de entrenamiento (Lee et al., 2012; Pourbasheer et al., 2014; Xing et al., 2014; Yousefinejad & Hemmateenejad, 2015). Como paso adicional, se puede considerar la construcción de modelos más robustos a partir del consenso de varios modelos (modelos bases). Estos modelos denominados de consenso o de "ensemble" parten de la hipótesis de que la conjunción de los modelos bases tiene mejor rendimiento que la predicción de los modelos bases por si solos (Khan et al., 2019; Martínez et al., 2018; Papa et al., 2014; K. Roy et al., 2018; K. Roy & Ambure, 2016; H. Sun et al., 2017).

Cualquier estudio QSAR debe conducir a construir modelos estadísticamente robustos, capaces de hacer predicciones precisas y fiables de las actividades biológicas en nuevos compuestos. El principio 4 de la OECD (OECD, s/f) enfatiza la necesidad de utilizar las medidas apropiadas para medir la bondad de ajuste, solidez y predictibilidad de los modelos construidos. En este sentido, existen diferentes estrategias para validar la calidad de los modelos construidos; ya sea utilizando el conjunto de entrenamiento o conjuntos de validación externos. Es importante señalar que para validar la calidad de los modelos en el conjunto de entrenamiento (validación interna) es necesario utilizar alguna técnica de validación como pueden ser: validación cruzada dejando a uno afuera (LOO), validación cruzada de N pliegues (CV), bootstrapping (Gramatica, 2007; Gramatica & Sangion, 2016; Yousefinejad & Hemmateenejad, 2015), entre otras (K. Roy et al., 2012; Todeschini et al., 2016). Para la validación externa se pueden tener diferentes criterios, tales como el error cuadrático medio (Consonni et al., 2010), área bajo la curva (Majumdar & Basak, 2018), correlación de Matthews (MCC) (K. Roy et al., 2015a), entre otros (Golbraikh et al., 2017; Gramatica, 2007; Gramatica & Sangion, 2016; Tropsha, 2010; Yousefinejad & Hemmateenejad, 2015).

El principio 3 de la OECD indica la necesidad de definir el dominio de aplicabilidad (AD por sus siglas en inglés) para los modelos construidos (Yousefinejad & Hemmateenejad, 2015). EL AD se construye a partir del conjunto de entrenamiento y define el espacio químico donde la predicción del modelo es confiable (Golbraikh et al., 2017; Gramatica, 2007; Gramatica & Sangion, 2016; Tropsha, 2010; Yousefinejad & Hemmateenejad, 2015). Lo anterior indica que si dado un nuevo compuesto se desea determinar si presenta o no una actividad determinada a partir de un modelo, es necesario primero determinar si el compuesto está dentro del AD del modelo. Si el compuesto está dentro del AD entonces la predicción del modelo es confiable.

Por último, la OECD en su principio 5 pide que de ser posible exista un mecanismo que permita interpretar el modelo construido (OECD, s/f). Este principio es opcional, dado que existen diversas técnicas de modelación o DMs que no pueden ser interpretados. Estos pasos básicos junto a los principios de la OECD han condicionado desde su creación el flujo básico necesario para el desarrollo de modelos QSAR (Mauri et al., 2016; K. Roy, 2020; K. Roy et al., 2015b; *The Validation of Alternative Test Methods*, 2019). De igual forma, ha propiciado el desarrollo de nuevas técnicas y la aplicación de nuevos algoritmos, métodos de modelado y prácticas de validación, que, aunque se definieron para la modelación QSAR, han sido aplicados en otras áreas de investigación (por ejemplo, ciencia de materiales, nanotecnología, biomateriales, entre otros) (Muratov et al., 2020).

2.3 Algoritmos genéticos

Los algoritmos genéticos (AG) constituyen la estrategia bioinspirada más conocida en la actualidad (Olaque, 2016). Los principios básicos de estos fueron establecidos por Holland (Holland, 1975), quien lo concibió como un medio para estudiar el proceso de adaptación de las especies con el fin de reproducir las características de los sistemas adaptativos naturales dentro del diseño de un sistema artificial. Los algoritmos genéticos clásicos se consideran técnicas de búsqueda guiada. En estos se define una población de posibles soluciones para resolver un problema de optimización. Cada solución se codifica utilizando un vector de longitud finita denominado cromosoma. La idea es emular diferentes mecanismos existentes en la naturaleza que promueven la evolución de las mejores soluciones (individuos). Estas soluciones están asociadas con un valor de aptitud que refleja qué tan buena es la solución en comparación con otras en la población. Por lo tanto, cuanto mayor sea el valor de aptitud de un individuo, mayores serán sus posibilidades de supervivencia.

El Algoritmo 1 muestra los pasos básicos de un algoritmo genético. Al inicio del algoritmo se crea una población inicial de posibles soluciones. Esta población será el punto de partida para la búsqueda. En las próximas iteraciones el objetivo será mejorar las posibles soluciones a partir de los operadores genéticos. Basándose en la evaluación de la aptitud de las soluciones, el operador de selección artificial, el cual está inspirado en la teoría de selección natural, escoge las soluciones a partir de las cuales una nueva población va a ser generada. El proceso de selección determina cuál es el individuo o los individuos con mayor probabilidad de sobrevivir. La hipótesis detrás hace referencia a que los individuos con mayor calidad tienen mayor probabilidad de sobrevivir y de generar descendencia.

Algoritmo 1. Algoritmo genético

```

1.  $gen \leftarrow 0$ 
2.  $initialize(pop(gen))$  // Comúnmente se inicializa de forma aleatoria
3.  $compute\_fitness(pop(gen))$ 
4. repeat
5.    $gen \leftarrow gen + 1$ 
6.    $parents \leftarrow selection(pop(gen - 1))$ 
7.    $offspring \leftarrow crossover(parents)$ 
8.    $offspring' \leftarrow mutation(offspring)$ 
9.    $pop(gen) \leftarrow replacement(pop(gen - 1), parents, offspring')$ 
10.   $compute\_fitness(pop(gen))$ 
11. until stop_condition

```

Una vez definidas las soluciones padres se aplican los diferentes operadores genéticos. Primero se aplica el operador de cruzamiento, el cual genera una nueva población a partir de la combinación de las

soluciones seleccionadas. Para luego aplicar el operador de mutación sobre las nuevas soluciones generadas, modificando partes de la estructura. Este operador tiene como objetivo generar variantes en la estructura genotípica de los individuos. El proceso se repite hasta que una condición de parada sea alcanzada.

2.3.1 Algoritmos coevolutivos

Asociado al concepto de evolución existe el concepto de evolución cooperativa. En el cual se hace referencia a que las especies en la naturaleza no evolucionan de forma independiente, sino que esta evolución depende además del ambiente en el cual se encuentra dicha especie y a otras especies que comparten el mismo ambiente. Un ejemplo clásico de esta cooperación es el caso de la evolución de las plantas terrestres y como estas dieron un salto relevante en el número de especies y población, así como la complejidad de sus estructuras una vez que aparecieron las plantas con flores y los insectos polinizadores (Leslie et al., 2021).

Basados en este concepto, Potter y De Jong proponen un enfoque coevolutivo cooperativo en 1994 (Potter & Jong, 2000, 1994). El objetivo principal fue proponer un nuevo enfoque que le pudiera dar frente a los cada vez más complejos problemas, tanto en estructura como en tamaño (Potter & Jong, 1994). Para esto se propuso un sistema que cumpliera con las siguientes ideas: 1) una especie representa un subcomponente de una solución potencial; 2) las soluciones completas se obtienen ensamblando miembros representativos de cada una de las especies presentes; 3) la asignación de créditos a nivel de especie se define en términos de la idoneidad de las soluciones completas en las que participan los miembros de la especie; 4) cuando sea necesario, el número de especies (subpoblaciones) debería evolucionar; y 5) la evolución de cada especie (subpoblación) es manejada por un AG estándar. Definiendo tales sistemas como algoritmos genéticos coevolutivos cooperativos (Potter & Jong, 1994). Entonces una definición simple de un algoritmo genético coevolutivo cooperativo (ACC), ver Algoritmo 2, puede venir dada por ser un algoritmo evolutivo (o una colección de estos) en el que la aptitud de un individuo de una especie depende de la relación entre ese individuo y otros individuos de otras especies (Wiegand & Jong, 2004). Por lo que un ACC debe cumplir con dos condiciones: 1) el algoritmo debe analizar al menos 2 poblaciones y 2) la función de calidad de un individuo depende de la relación entre este individuo y los individuos de otras especies (Wiegand & Jong, 2004).

Algoritmo 2. Algoritmo genético coevolutivo cooperativo

```

1.  $gen \leftarrow 0$ 
2. for  $s \leftarrow 1$  to  $size(species)$ 
3.    $initialize(pop_s(gen))$ 
4.    $offspring\_to\_share \leftarrow selection(pop_s(gen))$ 
5.    $share\_offspring\_to\_species(offspring\_to\_share, species)$ 
6.    $offspring\_coop \leftarrow recv\_for\_coop(species)$ 
7.    $compute\_fitness(pop_s(gen))$ 
8. end for
9. Repeat
10.   $gen \leftarrow gen + 1$ 
11.  for  $s \leftarrow 1$  to  $size(species)$ 
12.     $parents \leftarrow selection(pop_s(gen - 1))$ 
13.     $offspring \leftarrow crossover(parents)$ 
14.     $offspring' \leftarrow mutation(offspring)$ 
15.     $pop_s(gen) \leftarrow replacement(pop_s(gen - 1), parents, offspring')$ 
16.     $compute\_fitness(pop_s(gen))$ 
17.  end for
18. until stop\_condition

```

2.4 Medidas e integrales difusas

La agregación es un proceso en el análisis multi criterio donde el rendimiento de una alternativa dada (x_1, x_2, \dots, x_n) , con respecto a un conjunto de múltiples criterios (c_1, c_2, \dots, c_n) , se resume en un solo valor de calidad (Krishnan et al., 2017). Este proceso se repite para cada una de las alternativas involucradas en el análisis, para determinar luego, basadas en el valor de calidad la o las alternativas más favorables. A la función que sintetiza los puntajes de desempeño de una alternativa en un solo valor se suele denominar operador de agregación (Krishnan et al., 2017).

Los operadores más utilizados en la actualidad son los operadores aditivos, tipo suma ponderada o medias ponderadas (Grabisch & Labreuche, 2010). Estos operadores son efectivos en muchas aplicaciones, pero ninguno de estos operadores captura las interacciones que suelen existir entre los atributos de evaluación (Grabisch, 1995; Grabisch & Labreuche, 2005, 2010). Para ciertos problemas es necesario tener en cuenta no solo el aporte de los criterios de forma individual, sino además analizar como interactúan entre ellos. Un ejemplo de esto puede ser determinar el equipo de trabajadores idóneo, donde es prudente tener en cuenta qué tan bien trabajan de manera conjunta. En estos casos, las medidas difusas pueden ser más útiles.

2.4.1 Medidas difusas

Sea X un conjunto finito y $A \in \mathcal{p}(X)$, la función $\mu: A \rightarrow [0,1]$ se dice que es una medida difusa si cumple con las siguientes condiciones:

$$\mu(\emptyset) = 0 \quad (3)$$

$$\mu(X) = 1 \quad (4)$$

$$\mu(A) \geq 0 \quad (5)$$

$$\text{Si } A, B \in X \wedge A \subseteq B \text{ entonces } \mu(A) \leq \mu(B) \quad (6)$$

El valor $\mu(A)$ se puede considerar como el peso o grado de importancia de considerar los criterios incluidos en A . Para $A = \{x_i\}$, con $x_i \in X$, $\mu(A) = \mu(x_i)$, donde $\mu(x_i)$ se denomina densidad difusa o medida singleton (García-Jacas, Cabrera-Leyva, Marrero-Ponce, Suárez-Lezcano, Cortés-Guzmán, Pupo-Meriño, et al., 2018).

Estas condiciones establecen primero: los límites que deben cumplir la función mediante las ecuaciones 3, 4 y 5, donde se define que la medida para un subconjunto de criterios estará entre 0 y 1. Por otro lado la ecuación 6 define la propiedad de monotonía que deben cumplir todas las medidas difusas. Esta condición garantiza que la medida difusa de un conjunto no disminuye cuando nuevos criterios son agregados.

Según la forma en que se agregan los criterios, las medidas difusas pueden ser clasificadas en:

Medida aditiva: Una medida difusa μ es aditiva si para todo par de conjuntos disjuntos $A, B \in X$, se tiene que $\mu(A \cup B) = \mu(A) + \mu(B)$.

Medida sub aditiva: Una medida difusa μ es sub aditiva si para todo par de conjuntos disjuntos $A, B \in X$, se tiene que $\mu(A \cup B) \leq \mu(A) + \mu(B)$.

Medida súper aditiva: Una medida difusa μ es súper aditiva si para todo par de conjuntos disjuntos $A, B \in X$, se tiene que $\mu(A \cup B) \geq \mu(A) + \mu(B)$.

En la literatura hay reportadas varias medidas difusas, entre las cuales podemos encontrar Sugeno λ -measure (Beliakov et al., 2016), P-measure (Zadeh, 1978), L_m -measure (H.-C. Liu et al., 2007), δ -measure (Hsiang-Chuan Liu and Der-Bang Wu and Yu-Du Jheng and Tian-Wei Sheu, 2009), Q-measure (Mohamed & Weimin Xiao, 2003) entre otras. Las medidas λ -measure y P-measure son de las más utilizadas en la literatura. En el caso de P-measure, donde se devuelve solo el mayor valor de un conjunto de datos, tiene la desventaja de no ser lo suficientemente sensitiva. Por otro lado λ -measure necesita determinar el valor de λ a partir de un polinomio de orden N , siendo N el número de criterios de ecuaciones, lo que hace complejo su cálculo.

2.4.2 Integrales difusas

Definida una medida difusa y teniendo un conjunto de criterios, entonces la calidad de cada subconjunto de criterios puede ser definido. Teniendo esto, el próximo paso sería integrar la relevancia que tienen formar subconjuntos con ciertos criterios determinados con la evaluación de estos criterios, por separado, sobre el objeto al cual se le quiere determinar su calidad. A las funciones que realizan esta integración se les denomina integrales difusas (Beliakov et al., 2016; H.-C. Liu et al., 2007), siendo las más utilizadas la integral de Sugeno (Beliakov et al., 2016; Marichal, 2000) y la integral de Choquet (Beliakov et al., 2016; Choquet, 1954; Krishnan et al., 2017; H.-C. Liu et al., 2007).

La integral de Choquet fue introducida por Gustave Choquet en 1953 (Choquet, 1954) y desde entonces ha sido utilizada ampliamente, sobre todo en el campo de funciones multicriterio (Lust, 2015) y recientemente como método de agregación en la conformación de nuevos descriptores moleculares (García-Jacas, Cabrera-Leyva, Marrero-Ponce, Suárez-Lezcano, Cortés-Guzmán, Pupo-Meriño, et al., 2018). Formalmente, dado un conjunto finito de criterios $X = \{x_1, x_2, \dots, x_n\}$, las evaluaciones de los criterios en el objeto analizado $Y = \{y_1, y_2, \dots, y_n\}$ y una medida difusa μ definida sobre X , la integral de Choquet C de X respecto a μ se define como:

$$C_{\mu}(x_1, x_2, \dots, x_n) = \sum_{i=1}^N y_i [\mu(A_{(i)}) - \mu(A_{(i-1)})] \quad (7)$$

Donde $A_{(i)}$ denota una permutación de X ordenada según Y , de tal forma de que si $A_{(i)} = \{x_{(i)}, \dots, x_{(N)}\}$ entonces $y_i \geq y_{i+1} \geq \dots \geq y_n$ con $i \geq 1$ y $A_0 = \emptyset$. Como ejemplo supongamos que tenemos el conjunto

de cuatro criterios $X = \{x_1, x_2, x_3, x_4\}$, de tal forma que $y_4 \geq y_2 \geq y_3 \geq y_1$ la integral de Choquet se define entonces de la siguiente forma:

$$\begin{aligned}
 C_{\mu}(x_1, x_2, x_3, x_4) = & y_4[\mu(x_4, x_2, x_3, x_1) - \mu(x_2, x_3, x_1)] \\
 & + y_2[\mu(x_2, x_3, x_1) - \mu(x_3, x_1)] \\
 & + y_3[\mu(x_3, x_1) - \mu(x_1)] \\
 & + y_1[\mu(x_1)]
 \end{aligned} \tag{8}$$

2.5 Conclusiones parciales

Los descriptores moleculares son los encargados de codificar la información estructural de las moléculas a partir de una representación simbólica dada. En este sentido, se puede decir que un paso fundamental para la construcción de un modelo QSAR es determinar cuáles descriptores moleculares serán tenidos en cuenta. Una vez determinados los descriptores a considerar, el proceso de modelación es guiado por los principios definidos por la OECD. Estos principios definen los puntos necesarios para poder concluir un modelado QSAR exitoso y remarcan la necesidad de definir una actividad a predecir, un algoritmo de modelado no ambiguo, un dominio de aplicabilidad, una apropiada función de calidad para medir la calidad de los modelos construidos y un mecanismo de interpretación de los modelos si es posible.

Por otro lado, los algoritmos genéticos constituyen el algoritmo bioinspirado más conocido en la actualidad. Este algoritmo se considera una estrategia de búsqueda guiada, al reproducir los pasos básicos de la selección natural de Darwin: selección, cruzamiento, mutación y remplazo para la búsqueda de soluciones optimas o aproximadas a problemas complejos. Por otro lado, teniendo en cuenta el incremento de la complejidad de los problemas que son analizados con estrategias evolutivas, se propusieron los algoritmos genéticos coevolutivos. Estos parten del hecho de que las especies no evolucionan de forma independiente, sino que su evolución está condicionada al ambiente en el cual se desarrollan. Un algoritmo coevolutivo debe cumplir con dos condiciones: se deben analizar al menos dos poblaciones y la calidad de los individuos en una población dada debe ser dependiente de individuos en otras poblaciones. Estos son efectivos para abordar problemas de alta dimensionalidad debido a su capacidad de distribuir la búsqueda, reducir complejidad y aprovechar el paralelismo.

Capítulo 3. Algoritmo para Explorar y Optimizar el Espacio de Configuración del Descriptor

En este capítulo se presenta el algoritmo AExOp-DCS, el cual está basado en algoritmos genéticos. El algoritmo devuelve un subconjunto “óptimo” de DMs respecto a las moléculas bajo estudio y la actividad que se quiere modelar. Se describe, además, la función utilizada para determinar la calidad de los descriptores moleculares. Esta función determina la calidad de un DM mediante la agregación de cuatro criterios distintos. Por otro lado, se describe la función para determinar la calidad de un subconjunto de descriptores moleculares a partir de la correlación existente entre los DMs del subconjunto y de la correlación existente entre los descriptores y la clase a predecir. Finalmente, se realiza un análisis de la calidad de los subconjuntos encontrados a partir del rendimiento de los modelos que se construyen a partir de ellos.

3.1 Función para determinar la función de calidad de los descriptores

En esta sección se describen los criterios considerados para determinar la calidad de los DMs, así como el operador de agregación para determinar el peso total de los DMs a partir de los criterios utilizados. Los criterios son correlación de Pearson (R) (Kallner, 2018), entropía de Shannon (ES) (Kallner, 2018), *ReliefF* (Kohavi & John, 1997) y la disminución media de la impureza (MDI) (Loupe et al., 2013). El valor de la entropía de Shannon es una medida del contenido de información (Kallner, 2018), R es una estimación de la asociación lineal a la clase a predecir (Kallner, 2018), mientras que *ReliefF* estima la importancia de los atributos según qué tan bien sus valores distinguen entre instancias que están cercanas entre sí (Kohavi & John, 1997).

3.1.1 Métricas para determinar la calidad de los descriptores moleculares

Entropía de Shannon: Es un procedimiento no supervisado que mide el contenido de información que contiene una variable aleatoria (Barigye et al., 2013). Esta métrica ha sido ampliamente utilizada para estimar la calidad de descriptores moleculares, ya que es deseado que los DMs tengan valores elevados de entropía (García-Jacas et al., 2014, 2015, 2019; Urias et al., 2015). Esto se debe a que descriptores deseables para análisis quimio-métricos debieran tener valores elevados de entropía como un indicador

de su tendencia a cambiar gradualmente con la modificación de la estructura molecular; mientras que variables con poco contenido de información debieran tener valores bajos de entropía. La ES se define como:

$$ES = - \sum_{i=1}^n P(x_i) \log P(x_i) \quad (9)$$

donde n es el número de compuestos y x_i es el valor del descriptor para el compuesto i . Una entropía cercana a cero, indica pobre contenido de información. Por ejemplo, un descriptor que devuelve un valor constante, sin importar el compuesto, tiene entropía 0.

ReliefF: Es un algoritmo que caracteriza cada uno de los atributos en dependencia de cuán bien puedan diferenciar instancias que están cercas una de las otras (Kira & Rendell, 1992). El Algoritmo 3 muestra los pasos básicos de esta función. En el primer paso, se selecciona de forma aleatoria una instancia (R). A partir de esta instancia, *ReliefF* busca el vecino más cercano (H) con la misma clase de R , y el vecino más cercano (M) con clase diferente a R . A partir de estas tres instancias, *ReliefF* actualiza el peso de cada atributo siguiendo la idea de que: si las instancias R y H tienen valores diferentes para un DM i , entonces esto quiere decir que el atributo i separa dos instancias de la misma clase, evento no deseado por lo que se decrementa el peso de i . Por otro lado, si para las instancias R y M el atributo i tiene valores diferentes, esto implica que el atributo i separa dos instancias con clases diferentes, evento deseado, por lo que su peso aumenta. Este algoritmo se repite m veces, donde m es definido por el usuario. Esta estrategia se crea para la selección de variables en problemas de clasificación binaria, pero ha sido adaptada satisfactoriamente a problemas de regresión (Kohavi & John, 1997).

Correlación de Pearson: Es una medida estadística de dependencia lineal desarrollada por el estadístico Karl Pearson. Esta evalúa la relación y dirección de la asociación entre dos variables (Keying Ye et al., 2012). La correlación de Pearson (ecuación 9) ha sido ampliamente utilizada en la modelación QSAR para establecer relaciones cuantitativas entre los DMs y la actividad o propiedad a modelar (Gasteiger, 2003; Mauri et al., 2016; K. Roy, 2020).

Disminución media de la impureza (MDI): Indica cuánto contribuye un DM a la efectividad de un clasificador basado en *Random Forest* (Louppe et al., 2013). En este sentido, se puede definir la importancia de un descriptor molecular DM_m para predecir la actividad Y (ver ecuación 10) mediante la

suma de la disminución de la impureza $p(t)\Delta i(s_t, t)$ para todos los nodos t donde DM_m es utilizado. Esta suma se pondera respecto a los N_T árboles que conforman el bosque (Louppe et al., 2013)).

Algoritmo 3. Algoritmo para determinar la calidad según *ReliefF* para cada uno de los DMs

Entrada

- **Matriz de tamaño $M \times (D + 1)$**
 /* donde se tiene la evaluación de los D DMs en las M moléculas */
 /* En la columna extra se tiene la actividad de cada un de las moléculas */
-

Salida

- **Vector w con los pesos de cada descriptor**
-

1. **for $i = 1$ to D**
 2. $w[i] \leftarrow 0$ // inicializar los pesos en 0
 3. **end for**
 4. **for $i = 1$ to m**
 5. **Escoger aleatoriamente una instancia R**
 6. **Encontrar el vecino más cercano a R con la misma clase de R (H)**
 7. **Encontrar el vecino más cercano a R con clase diferente a R .**
 8. **for $i = 1$ to D**
 9. $w(i) \leftarrow w(i) - \frac{\text{diff}(i,R,H)}{m} + \frac{\text{diff}(i,R,M)}{m}$
 10. **end for**
 11. **end for**
 12. **return w**
-

$$r_{jk} = \frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 (x_{ik} - \bar{x}_k)^2}} \quad (9)$$

$$MDI(DM_m) = \frac{1}{N_T} \sum_t \sum_{t \in T, v(s_t)=DM_m} p(t)\Delta i(s_t, t) \quad (100)$$

donde $p(t) = \frac{N_t}{N}$, la proporción de objetos que llegan al nodo t respecto al total de objetos N , $v(s_t)$ es la variable utilizada para dividir el nodo t y $\Delta i(s_t, t)$ define la impureza generada en el nodo t a partir de utilizar s_t en la división del nodo (ver Ecuación 11) a partir de cierta métrica de impureza i .

$$\Delta i(s_t, t) = i(t) - \frac{N_{t_l}}{N_t} i(t_l) - \frac{N_{t_r}}{N_t} i(t_r) \quad (11)$$

MDI está definido para cualquier función de impureza (Louppe et al., 2013), siendo las más comunes el índice de Gini, Entropía de Shannon o la estimación de la varianza (Louppe et al., 2013).

Definida MDI, se puede definir entonces el concepto de variable **irrelevante** asociada a dicha métrica. Sea X_m una variable a analizar, X_m es **irrelevante** para Y si y solo si su importancia con un tamaño de muestra infinito, calculada con un conjunto infinito de árboles totalmente desarrollados y contruidos de forma aleatoria para predecir Y , es igual a 0. De forma contraria, se puede definir el concepto de variable **relevante**, siendo aquellas variables que no son irrelevantes. Ambos conceptos, variable irrelevante y variable relevante dado un bosque de árboles serán utilizados en próximas secciones.

3.1.2 Operador de agregación basada en la Integral de Choquet para determinar la calidad global de los descriptores.

Para determinar la calidad global de los DMs se define una función multicriterio basada en la integral de Choquet (Lust, 2015; Murofushi & Sugeno, 1991). Los cuatro criterios que se consideran son correlación de Pearson (R) (Kallner, 2018), entropía de Shannon (ES) (Kallner, 2018), *ReliefF* (RE) (Kohavi & John, 1997) y la disminución media de la impureza (MDI) (Louppe et al., 2013). Los valores de *ReliefF*, correlación y MDI fueron calculados utilizando el API de la versión 3.9 de Weka (*WEKA software, s/f*).

Los criterios se fusionan mediante la integral de Choquet con el objetivo de tener en cuenta la interrelación existente entre ellos, calculada mediante el uso de la medida difusa no aditiva Q-measure (Mohamed & Weimin Xiao, 2003). Por lo que, dado $X = \{ES, RE, MDI, R\}$, el conjunto de criterios a considerar, y $P(X)$ el conjunto potencia de X [$|P(X)| = 2^{|X|}$], la medida difusa Q-measure sobre $A \in P(X)$: $q(A) \rightarrow [0,1]$ se define como:

$$q(A) = \frac{\prod_{x_i \in A} (1 + \lambda \mu(x_i)) - 1}{\prod_{x_i \in X} (1 + \lambda \mu(x_i)) - 1}, \lambda \neq 0 \quad (12)$$

donde $\mu(x_i)$ es el valor de densidad difusa para el criterio x_i , y $\lambda \in [-1, \infty]$ el grado de interacción a considerar. Los valores de densidad difusa se pueden interpretar como los valores de peso que se utilizan tradicionales (valores de importancia). En este trabajo se toman como valores difusos: $\mu(R) = 0.05$, $\mu(ES) = 0.1$, y $\mu(MDI) = \mu(RE) = 0.3$, mientras que $\lambda = 0.5$. Definida la medida y los valores difusos, el operador de agregación basada en la integral de Choquet (C_q) respecto a la medida difusa Q-measure (q) se define como:

$$C_q(X) = \sum_{i=1}^{|X|} y_i [q(A_{(i)}) - q(A_{(i-1)})] \quad (13)$$

donde y_i es la evaluación del criterio x_i para el DM y $A_{(i)}$ denota una permutación de X ordenada según Y , de tal forma de que si $A_{(i)} = \{x_{(i)}, \dots, x_{(N)}\}$ entonces $y_i \geq y_{i+1} \geq \dots \geq y_n$ con $i \geq 1$ y $A_0 = \emptyset$.

Como ejemplo, supongamos que dado un DM d , se tienen los siguientes criterios: $R(d) = 0.97$, $ES(d) = 0.75$, $MDI(d) = 0.15$ y $RE(d) = 0.63$. A se define entonces como $A = \{R, ES, RE, MDI\}$ y la integral de Choquet respecto a Q -measure (C_q) se desarrolla como:

$$\begin{aligned} C_q(R, ES, RE, MDI) &= R(d)[q(R, ES, RE, MDI) - q(ES, RE, MDI)] \\ &+ ES(d)[q(ES, RE, MDI) - q(RE, MDI)] \\ &+ RE(d)[q(RE, MDI) - q(MDI)] \\ &+ MDI(d)[q(MDI)] \end{aligned} \quad (14)$$

Sustituyendo se tiene entonces:

$$\begin{aligned} C_q(R, ES, RE, MDI) &= 0.97[1.00 - 0.92] \\ &+ 0.75[0.92 - 0.76] \\ &+ 0.63[0.76 - 0.35] \\ &+ 0.15[0.35] \\ &= 0.5084 \end{aligned} \quad (15)$$

3.2 Función para determinar la calidad de los subconjuntos

Un paso importante en el algoritmo que se está proponiendo en este trabajo es determinar el subconjunto que contiene los mejores DMs dado una población total de DMs. Para esto, se va a utilizar el algoritmo de selección de rasgos denominado selección de características basado en la correlación (CFS por sus siglas en inglés) (Hall, 2000). Este algoritmo clasifica los subconjuntos evaluados acorde a una función de calidad de subconjuntos basada en correlación, partiendo de la base de que un buen subconjunto de atributos es aquel donde los atributos que lo componen están altamente correlacionados con la clase y a la misma vez tienen poca correlación entre ellos.

En esta función, atributos irrelevantes deben ignorarse porque tendrán una baja correlación con la clase. Los rasgos redundantes deben descartarse de igual forma porque estarán altamente correlacionados con uno o más de los rasgos restantes. Para que un rasgo sea tomado en cuenta, este debe caracterizar la variable a predecir en un espacio determinado que no es caracterizado por otro rasgo (Hall, 2000). CFS está definida por la siguiente ecuación:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (16)$$

donde M_S es la calidad o mérito del subconjunto S , con $|M_S| = k$, $\overline{r_{cf}}$ es la media de la correlación entre los rasgos $f \in S$ y la clase a predecir, mientras que $\overline{r_{ff}}$ es la media de la correlación existente entre los atributos de S . Para guiar la búsqueda se utiliza el algoritmo de búsqueda conocido como Primero el mejor (Hall, 2000).

3.3 AExOp-DCS: Algoritmo para Explorar y Optimizar el Espacio de Configuración del Descriptor

Como se menciona al inicio del capítulo, el algoritmo AExOp-DCS está basado en un algoritmo genético. La estrategia tiene como objetivo explorar y optimizar el DCS de cada algoritmo de cálculo de DM que se está considerando para resolver el problema. Como se describe en el capítulo anterior, la representación cromosómica, los operadores genéticos y la función de aptitud son componentes clave de un AG. Un AG clásico codifica las soluciones candidatas en vectores de longitud finita denominados cromosomas. Cada uno de ellos está compuesto por un conjunto de parámetros (genes) cuyos valores (alelos) pueden pertenecer a diferentes dominios. Cuando comienza el AG, primero se construye una población inicial con un cierto número de cromosomas. Luego, se calcula el fitness de cada cromosoma para seleccionar los más adecuados para la recombinación, con la esperanza de obtener nuevas y mejores soluciones (descendencia). Como suele pasar en la naturaleza, en ocasiones la descendencia sufre algunas alteraciones denominadas mutaciones, de igual forma en un AG hay una probabilidad de que individuos dentro de la nueva población sufran algunas mutaciones. Estos pasos se repiten hasta que se cumple una condición de parada. El cromosoma con la aptitud más alta es la salida del AG.

Sin embargo, el enfoque propuesto difiere de un AG clásico principalmente porque cada cromosoma

(genotipo) representará un DM, y dado que no existe un solo DM capaz de codificar toda la información química existente en diferentes compuestos, la solución que se obtendrá es un conjunto de cromosomas en lugar del mejor. Este algoritmo se basa entonces en la evolución de las poblaciones de DCS. Por tanto, cada población tendrá cromosomas de tamaño n , cuyas posiciones (genes) corresponderán a los n parámetros de entrada de un algoritmo de cálculo de DM, por ejemplo, RDF (Hemmer et al., 1999), 3D-MoRSE (Devinyak et al., 2014), WHIM (Gramatica, 2006), entre otros (Mauri et al., 2016). El valor (alelo) que puede tomar cada gen dependerá del dominio de valores correspondiente al parámetro que representa. Ver las Figuras 4 y 5 para una mejor comprensión. En estas figuras se muestran las definiciones de los cromosomas para dos DCS distintos, la Figura 4A muestra la definición del cromosoma para los DMs RDF, mientras que la Figura 5A muestra la definición del cromosoma para los DMs 3D-MoRSE. Además, se muestran ejemplos de poblaciones para ambos algoritmos en las Figuras 4B y 5B, para finalmente mostrar cómo quedan las poblaciones luego de calcular los DMs representados en las poblaciones sobre el conjunto de moléculas (ver figuras 4C y 5C).

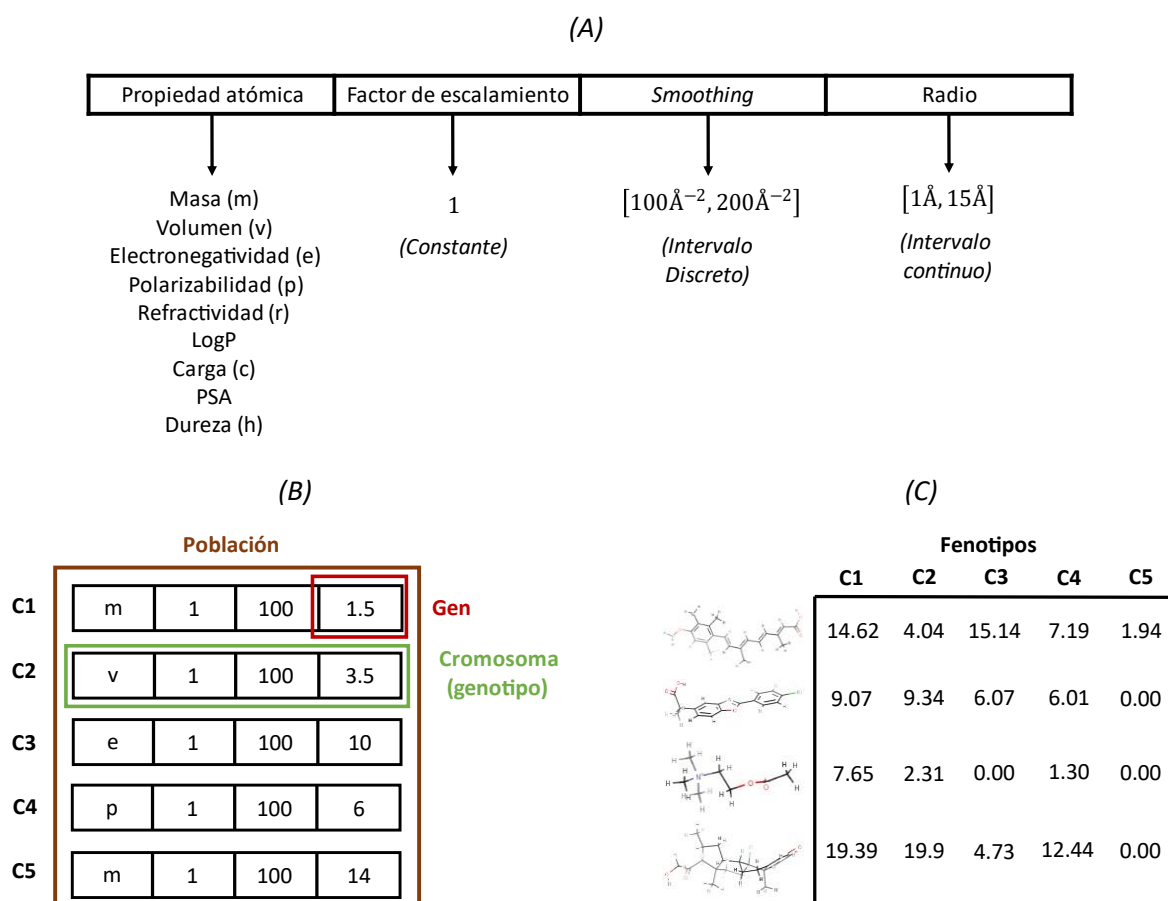


Figura 4. Definición del cromosoma para representar los DMs RDF. (A) Representación del cromosoma, (B) Ejemplo de una población con cromosomas que representan DMs RDF, (C) Evaluaciones de los cromosomas dentro de la población en un conjunto de moléculas de ejemplo.

Como las poblaciones serán diferentes entre sí porque representan diferentes DCS, los cromosomas entre poblaciones no se pueden recombinar. Por ejemplo, no se pueden recombinar los cromosomas que representan DCS correspondientes a los algoritmos RDF y 3D-MoRSE. Por lo tanto, el proceso evolutivo solo se realizará en cada población de forma independiente, pero se considerarán todas las poblaciones para determinar el mejor juego de cromosomas.

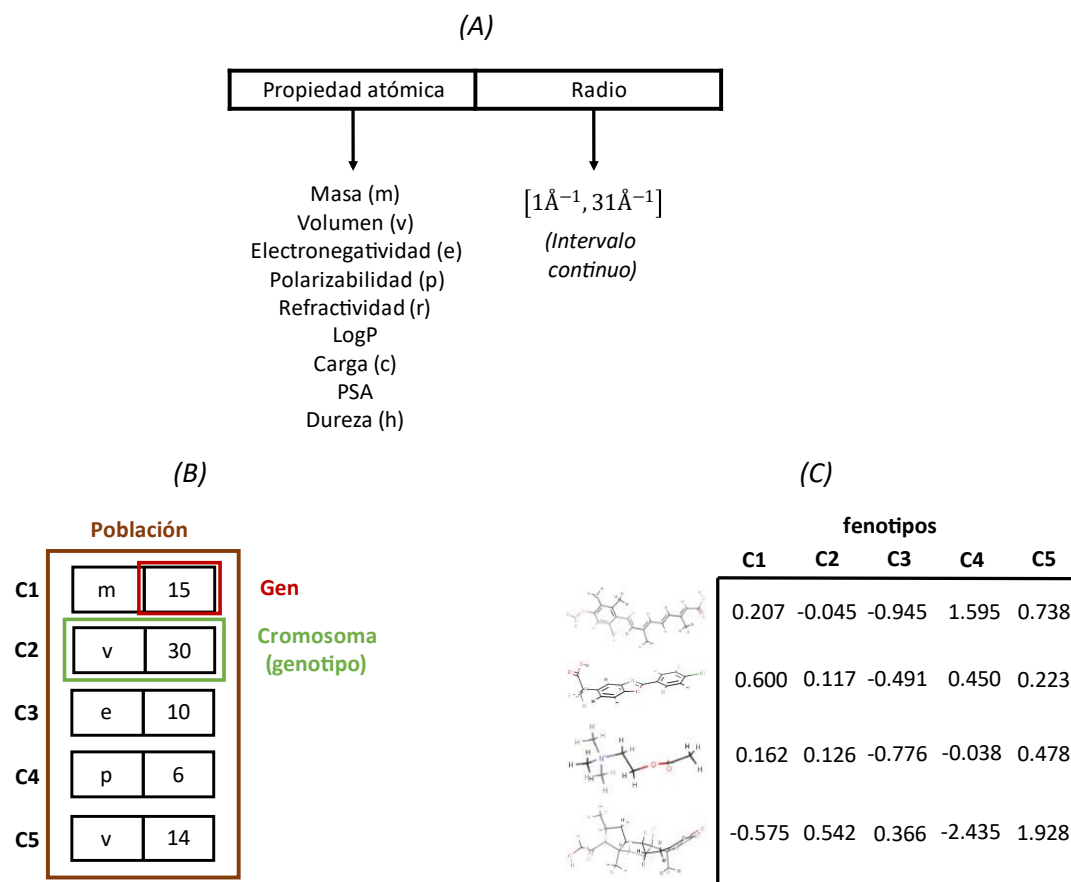


Figura 5. Definición del cromosoma para representar los DMs 3D-MoRSE. (A) Representación del cromosoma, (B) Ejemplo de una población con cromosomas que representan DMs 3D-MoRSE, (C) Evaluaciones de los cromosomas dentro de la población en un conjunto de moléculas de ejemplo.

Para calcular los valores de aptitud de los cromosomas (genotipos), es necesario calcular sus fenotipos en el conjunto de datos químicos que se va a analizar. Es decir, como cada cromosoma representa un DM, su valor numérico debe calcularse en cada compuesto bajo análisis. De esta forma, se determina una matriz de fenotipos de dimensión $C * P$ para cada población, donde C es el número de cromosomas y P es el número de moléculas en el conjunto de datos químicos bajo análisis. Las Figuras 4C y 5C muestran ejemplos de matriz de fenotipos correspondientes a las poblaciones representadas en las

Figuras 4B y 5B. Teniendo en cuenta que no existe una única métrica para evaluar la calidad de los DMs, una vez calculadas las matrices, los valores de aptitud de los DMs se determina utilizando un enfoque multi criterio. Para esto, se consideraron cuatro criterios de importancia: un criterio no supervisado basado en la Entropía de Shannon (Barigye et al., 2013) y tres criterios supervisados basados en las métricas Relief-F (Urbanowicz et al., 2018), Correlación de Pearson y *mean impurity decrease* (MDI) (Loupe et al., 2013). Después de calcular estos criterios para todos los DMs (representados como cromosomas), se calcula una aptitud global para cada uno de ellos agregando los cuatro criterios calculados a través de la integral de Choquet (Choquet, 1954; Murofushi & Sugeno, 1991), como se explica en la sección 3.1.

El Algoritmo 4 muestra el flujo básico de la estrategia AExOp-DCS. El algoritmo recibe como entrada al conjunto de moléculas de entrenamiento, la lista de DCSs sobre los cuales se buscarán los mejores DMs y la actividad o propiedad de las moléculas que se quiere modelar. Inicialmente se generan de forma aleatoria poblaciones por cada uno de los espacios de configuración de descriptores que se están analizando. Una vez definidas las poblaciones se determinan las matrices fenotípicas (ver ejemplos en la Figura 4 y Figura 5) con el objetivo de determinar la calidad de los descriptores representados en ellas (pasos 10 y 11 del algoritmo). Recordemos que la calidad de los descriptores se determina a partir de la integral de Choquet.

Después de obtener la calidad de los DMs se crea un *pool* donde estarán contenidos los fenotipos de cada una de las poblaciones analizadas. A este *pool* también se le agrega los fenotipos incluidos en el mejor subconjunto de DMs encontrados hasta el momento (línea 13 del algoritmo). El mejor subconjunto al inicio de la ejecución está vacío. Sobre el *pool* de fenotipos se ejecuta un algoritmo de selección de características para obtener el mejor conjunto de DMs respecto al conjunto de datos químicos y la actividad biológica bajo estudio (línea 14 del algoritmo). El método de selección va a estar basado en la estrategia de búsqueda CFS (ver sección 3.2). El valor devuelto (calidad del conjunto) por el método CFS es el valor de aptitud del algoritmo propuesto (denominado AExOp-DCS). Una vez determinado el mejor subconjunto dentro del *pool* se verifica que la calidad del nuevo subconjunto encontrado sea superior al mejor subconjunto actual. Si esto sucede el mejor subconjunto se actualiza con el nuevo subconjunto encontrado (ver líneas 15-18).

Posteriormente se pasa a generar una nueva población para cada uno de los DCS analizados (ver líneas 19-30). Para generar una nueva población se tiene dos formas: (1) se reinician las poblaciones generando cromosomas de forma aleatoria, o (2) se ejecutan los operadores genéticos. Para reiniciar las

poblaciones se necesita cumplir con un número determinado de iteraciones. Reiniciando las poblaciones se puede aumentar la diversidad de la búsqueda.

Algoritmo 4. Seudo código del algoritmo AExOp-DCS.

Entrada

- *Dataset* ← conjunto de compuestos de entrenamiento
 - *Activity* ← actividad/propiedad biológica de los compuestos
 - *DCSs* ← lista de los DCSs
-

Salida

- *Subconjunto best_subset con los mejores DMs encontrados*
-

```

1.  $t \leftarrow 0$ 
2.  $best\_fitness \leftarrow 0$ 
3.  $best\_solution \leftarrow empty$ 
4.  $populations \leftarrow size(DCSs)$ 
5. for  $i \leftarrow 1$  to  $size(DCSs)$ 
6.    $populations_{i,0} \leftarrow initialize(DCSs_i)$  /* inicializar de forma aleatoria /
7. end for
8. while  $t \leq T$ 
9.   for  $i \leftarrow 1$  to  $size(populations)$ 
10.     $phenotypes \leftarrow compute\_descriptors(populations_{i,t}, Dataset)$ 
11.     $fitness\_values \leftarrow compute\_chromosome\_fitness(phenotypes, Activity)$ 
12.  end for
13.   $merged\_pool \leftarrow join(phenotypes, best\_solution)$ 
14.   $temp\_best\_solution, temp\_best\_fitness \leftarrow CFS\_method(merged\_pool, Activity)$ 
15.  if  $temp\_best\_fitness > best\_fitness$ 
16.     $best\_fitness \leftarrow temp\_best\_fitness$ 
17.     $best\_solution \leftarrow temp\_best\_solution$ 
18.  end if
19.  if  $reset\_populations$ 
20.    for  $i \leftarrow 1$  to  $size(DCSs)$ 
21.       $populations_{i,t+1} \leftarrow initialize(DCSs_i)$ 
22.    end for
23.  else
24.    for  $i \leftarrow 1$  to  $size(DCSs)$ 
25.       $parents \leftarrow tournament\_selection(populations_{i,t}, fitness\_values)$ 
26.       $offspring \leftarrow HUX\_crossover(parents)$ 
27.       $offspring' \leftarrow mutation(offspring)$ 
28.       $populations_{i,t+1} \leftarrow replacement(populations_{i,t}, parents, offspring')$ 
29.    end for
30.  end if
31.   $t \leftarrow t + 1$ 
32. end while
33. return  $best\_solution$ 

```

Si la condición de reinicio no se cumple, entonces la próxima generación se va a generar ejecutando los operadores genéticos. Para este fin, se seleccionan varios pares de cromosomas desde cada población

(parámetro definido por el usuario) de acuerdo con su aptitud a través de la selección por torneo (Lavinás et al., 2018). Los pares de cromosomas seleccionados se recombinan a través del operador de cruce Uniforme medio (HUX por sus siglas en inglés) (Eshelman, 1991) para obtener nuevos cromosomas, es decir, mejores descriptores moleculares (DMs). Cada nuevo cromosoma podría mutarse seleccionando aleatoriamente un gen y cambiando su valor sobre su dominio de valores (mutación por scramble (Katoch et al., 2021)). Los nuevos cromosomas se ubican en la siguiente población utilizando la estrategia de reemplazo steady-state-no-duplicate (Burke & Kendall, 2014), donde los cromosomas a reemplazar son los cromosomas padres que se seleccionaron mediante torneo [35]. El algoritmo tiene como condición de parada el número de iteraciones.

3.4 Validación del algoritmo AExOp-DCS en la construcción de modelos QSAR a partir de los subconjuntos retornados.

3.4.1 Bases para la predicción de actividad biológica

Catorce conjuntos de estructuras químicas fueron utilizados para evaluar la calidad de los DMs devueltos por el algoritmo AExOp-DCS. La calidad de los subconjuntos se determinó mediante el rendimiento de los modelos QSAR construidos a partir de ellos para la predicción de diferentes actividades biológicas.

Ocho de los catorce conjuntos químicos (ver **¡Error! No se encuentra el origen de la referencia.**) se han utilizado ampliamente para evaluar diferentes enfoques QSAR (Ansari & Palmer, 2018; Bonachéra & Horvath, 2008; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; Klamt et al., 2012; Manchester & Czermiński, 2008; Martínez-Santiago et al., 2017; Sutherland et al., 2004; Tosco & Balle, 2012). Estos conjuntos se propusieron por Sutherland (Sutherland et al., 2004) y están compuestos por 114 inhibidores de la enzima convertidora de angiotensina (ECA), 111 inhibidores de la acetilcolinesterasa (ACHE), 163 ligandos para el receptor de benzodiazepina (BZR), 322 inhibidores de la ciclooxigenasa-2 (COX2), 397 inhibidores del dihidrofolato reductasa (DHFR), 66 inhibidores de glucógeno fosforilasa b (GPB), 76 inhibidores de termolisina (THER) y 88 inhibidores de trombina (THR).

Los otros seis conjuntos (Lavado et al., 2021; T. Liu et al., 2020; Nath et al., 2022) (ver **Tabla 2**) están relacionados con actividades ecotoxicológicas. Por un lado, cuatro de estos conjuntos se han utilizado para

predecir los efectos adversos de los productos químicos en un medio acuático (T. Liu et al., 2020). Estos conjuntos se componen de 194, 68, 149 y 143 compuestos orgánicos que se expusieron a 48, 96, 120 y 132 horas después de la fertilización, utilizando el embrión de pez cebra para medir los datos experimentales de LC_{50} . Por otro lado, también utilizamos un conjunto compuesto por 126 moléculas orgánicas volátiles (Nath et al., 2022) que se utilizaron para medir toxicidad por inhalación, definiendo como actividad a predecir la "concentración sin efecto adverso observado" (NOAEC). Finalmente, utilizamos un conjunto de datos para predecir los efectos adversos de los productos químicos en ecosistemas de agua dulce (Lavado et al., 2021). Este conjunto de datos está compuesto por 72 compuestos cuyos valores experimentales de LC_{50} se obtuvieron después de 24 horas de exposición al crustáceo de agua dulce *Thamnocephalus platyurus*.

Tabla 1. Descripción de los ocho conjuntos de compuestos químicos propuestos por Sutherland.

	ACE	ACHE	BZR	COX2	DHFR	GPB	THER	THR
Total	114	111	163	322	397	66	76	88
Entrenamiento	76	74	98	188	237	44	51	59
Prueba	38	37	49	94	124	22	25	29
Inactivo			16	40	36			
Actividad	pIC ₅₀	pIC ₅₀	pIC ₅₀	pIC ₅₀	pIC ₅₀	pK _i	pK _i	pK _i
# DMs*	6	8	9	9	9	8	7	9
Rango**	2.1-9.9	4.3-9.5	5.5-8.9	4.0-9.0	3.3-9.8	1.3-6.8	0.5-10.2	4.4-8.5

*: cantidad de DMs por modelo

*: rango de los valores de actividad

Tabla 2. Descripción de los seis conjuntos de compuestos químicos relacionados con actividad ecotoxicológica

	Toxicidad aguda <i>Zebrafish Embryo</i>				Toxicidad por inhalación	Toxicidad aguda <i>T. platyurus</i>
	48*	96*	120*	132*		24*
Total	194	68	149	143	126	72
Entrenamiento	155	54	119	114	97	56
Prueba	39	14	30	29	29	16
Actividad	pLC ₅₀	pLC ₅₀	pLC ₅₀	pLC ₅₀	pNOAEC	pLC ₅₀
# de DMs	21	12	21	21	9	9
Rango**	-2.8 - 4.4	-2.6 - 4.7	-2.5 - 4.1	1.1 - 5.2	2.3 - 9.21	-0.92 - 3.14

*: tiempo total de exposición (hora)

** : cantidad de DMs por modelo

***: rango de los valores de actividad

3.4.2 Procedimiento para validar la calidad de los DMs retornados

El algoritmo propuesto fue ejecutado en cada conjunto de entrenamiento para obtener el subconjunto “óptimo” de DMs respecto al conjunto de moléculas y a la actividad que se quiere predecir. A partir de los subconjuntos retornados se construyeron diferentes modelos y se comparó el rendimiento de estos respecto a varios modelos reportados en la literatura (Ansari & Palmer, 2018; Bonachéra & Horvath, 2008; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; Gupta et al., 2016; Hinselmann et al., 2011; Klamt et al., 2012; Lavado et al., 2021; T. Liu et al., 2020; Manchester & Czermiński, 2008; Martínez-Santiago et al., 2017; Nath et al., 2022; Sutherland et al., 2004; Tosco & Balle, 2012). La configuración utilizada para la ejecución de ambas estrategias es descrita en la Tabla 3.

Para garantizar la comparabilidad de los resultados con respecto a los modelos reportados en la literatura se recrearon los mismos escenarios en los que estos fueron construidos. En el caso de los modelos construidos a partir de los conjuntos reportados por Sutherland (Sutherland et al., 2004), la estrategia de modelación fue similar a la estrategia reportada en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020), donde se reportan los mejores modelos para estos conjuntos hasta la fecha. En este caso, los modelos se construyeron utilizando la técnica estadística de regresión lineal múltiple (MLR). Además, los modelos se construyeron a partir de la misma cantidad de DMs que la utilizada para construir los modelos en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020). Para cada uno de los conjuntos de Sutherland (Sutherland et al., 2004) se escogieron los mejores 30 modelos construidos a partir de cada uno de los subconjuntos devueltos y el rendimiento del mejor y peor modelo, así como el promedio de los modelos construidos, se compararon con respecto a los modelos reportados.

Para escoger los mejores modelos se tuvo en cuenta que la diferencia entre los rendimientos en conjunto de entrenamiento y prueba no fuera mayor a 0.05 para garantizar un adecuado balance bias-varianza. El rendimiento de los modelos en los conjuntos de entrenamiento y prueba se midió utilizando el coeficiente de determinación con validación cruzada a partir de 10 pliegues (R_{10-cv}^2) y validación externa (Q_{ext}^2), respectivamente. Además, para los seis conjuntos relacionados con las actividades ecotoxicológicas, se seleccionaron todos los modelos con valores R_{10-cv}^2 y Q_{ext}^2 mayores que los mejores valores reportados en la literatura. Estos modelos de igual forma se construyeron utilizando MLR, utilizando la misma cantidad de DMs utilizada para construir los modelos en (Lavado et al., 2021; T. Liu et al., 2020; Nath et al., 2022).

Tabla 3. Configuración utilizada por ambos algoritmos para determinar el subconjunto óptimo de DMs.

Parámetros	Valores
Numero de iteraciones	10,000
Dimensión de las poblaciones	100
Función para determinar la calidad de los subconjuntos	CFS
Condición de reinicio	Cada 2,000 iteraciones
Familia de DMs analizadas	QuBiLS-MAS (MASL: linear, MASQ: cuadrático, MASB: bilineal)
Operador de selección	Torneo k=5
Operador de cruzamiento	Medio uniforme (HUX) Probabilidad = 0.9
Operador de mutación	<i>scramble</i> Probabilidad = 0.1
Operador de remplazo	<i>steady-state-no-duplicate</i>
Función para determinar la calidad de los DMs	Integral de Choquet Medida difusa: Q-measure Valor lambda: 0.5 Peso de MDI: 0.3 Peso de Relief-F: 0.3 Peso de la Entropía de Shannon: 0.1 Peso de la correlación Person: 0.05
Filtros	Entropía Shannon > 0.1

3.4.3 Rendimiento de los modelos obtenidos

Las Tablas 4 y 5 muestran los valores mejor, peor y en promedio de R^2_{10-cv} obtenidos por los 30 mejores modelos construidos para los catorce conjuntos de datos, así como el rendimiento de varios modelos reportados en la literatura (Ansari & Palmer, 2018; Bonachéra & Horvath, 2008; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; Klamt et al., 2012; Lavado et al., 2021; T. Liu et al., 2020; Manchester & Czermiński, 2008; Martínez-Santiago et al., 2017; Nath et al., 2022; Sutherland et al., 2004; Tosco & Balle, 2012). En el caso de los modelos reportados en la literatura, se puede observar que estos reportan los valores de entrenamiento utilizando diferentes técnicas de validación, tales como, dejando uno afuera (R^2_{100}), validación cruzada con 5 pliegues (R^2_{5-cv}) o el coeficiente de determinación sin validación cruzada (R^2).

Tabla 4. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos construidos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de Sutherland.

	ACE*	ACHE	BZR	COX2	DHFR	GPB	THER	THR
#mejor QuBiLS-MAS ^a	<u>0.83</u>	<u>0.78</u>	<u>0.73</u>	<u>0.70</u>	<u>0.77</u>	<u>0.83</u>	<u>0.76</u>	<u>0.87</u>
##menor QuBiLS-MAS ^a	0.82	0.73	0.69	0.69	0.76	0.79	0.74	0.83
###promedio QuBiLS-MAS ^a	0.83 (.004)**	0.74 (.012)	0.70 (.008)	0.69 (.003)	0.76 (.002)	0.81 (.010)	0.75 (.006)	0.85 (.008)
QuBiLS-MIDAS (SI11 en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020)) ^b	0.75	<u>0.66</u>	<u>0.71</u>	0.63	0.7	<u>0.83</u>	<u>0.75</u>	0.81
GDIs (Tabla 3 en (García-Jacas et al., 2019)) ^b	<u>0.82</u>	<u>0.78</u>	0.7	<u>0.67</u>	<u>0.72</u>	<u>0.83</u>	0.74	<u>0.89</u>
CoMFA (Tabla 4 en (Sutherland et al., 2004)) ^b	0.68	0.52	0.32	0.49	0.65	0.42	0.52	0.59
COMSIA basic (Tabla 4 en (Sutherland et al., 2004)) ^b	0.65	0.48	0.41	0.43	0.63	0.43	0.54	0.62
COMSIA extra (Tabla 4 en (Sutherland et al., 2004)) ^b	0.66	0.49	0.45	0.57	0.65	0.61	0.51	0.72
EVA (Tabla 4 en (Sutherland et al., 2004)) ^b	0.70	0.42	0.40	0.45	0.64	0.58	0.48	0.47
HQSAR (Tabla 4 en (Sutherland et al., 2004)) ^b	0.72	0.34	0.42	0.50	0.69	<u>0.66</u>	0.49	0.50
2D (Tabla 4 en (Sutherland et al., 2004)) ^b	0.68	0.32	0.36	0.49	0.51	0.31	0.62	0.62
2.5D (Tabla 4 en (Sutherland et al., 2004)) ^b	0.72	0.31	0.35	0.55	0.53	0.46	0.66	0.52
SAMFA-RF (Tabla 2 en (Manchester & Czermiński, 2008)) ^b	0.69	0.58	0.43	0.38	0.70	<u>0.66</u>	0.52	0.53
SAMFA-SVM (Tabla 2 en (Manchester & Czermiński, 2008)) ^b	0.52	0.29	0.38	0.39	0.57	0.53	0.18	0.39
SAMFA-PLS (Tabla 2 en (Manchester & Czermiński, 2008)) ^b	0.65	0.54	0.49	0.40	0.68	0.61	0.60	0.56
jCompoundMapper (Tabla 6 en (Hinselmann et al., 2011)) ^b	0.69	0.57	0.56	0.55	<u>0.76</u>	0.53	0.53	0.58
O3Q (Tabla 1 en (Tosco & Balle, 2012)) ^b	0.69	0.52	0.42	0.48	0.70	0.55	0.48	0.59
O3QMFA (Tabla 1 en (Klamt et al., 2012)) ^b	0.65	0.41	0.41	0.43	0.69	0.30	0.47	0.65
O3A/O3Q (Tabla 1 en (Tosco & Balle, 2012)) ^b	0.71	0.55	0.46	0.46	0.66	0.50	0.67	0.68
COSMOsar3D (Tabla 1 en (Klamt et al., 2012)) ^b	0.71	0.53	0.45	0.54	0.69	0.61	0.58	0.74

^a: validación cruzada con 10 pliegues (R_{10-cv}^2)

^b: validación dejando uno fuera (R_{100}^2)

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

Si el rendimiento de los modelos construidos es analizado y se comparan con el rendimiento de los modelos ya reportados en la literatura, se puede observar cómo el rendimiento (R_{10-cv}^2) de los 30 modelos fue superior (o igual) que el rendimiento (R_{100}^2) del mejor modelo reportado en la literatura para los

conjuntos ACE, COX2 y DHFR. De igual forma se puede observar que para los conjuntos de datos ACHE, BZR, GPB y THER, el modelo con mayor rendimiento en el conjunto de entrenamiento (R_{10-cv}^2) fue superior (o similar) a los mejores modelos reportados en la literatura. Para estos conjuntos los modelos tienen un rendimiento promedio ligeramente inferior. Solo no se obtuvo un modelo con rendimiento superior al mejor modelo reportado en la literatura para el conjunto de moléculas THR, aunque todos tienen rendimiento superior al resto de los modelos reportados. Es importante resaltar que los modelos reportados en la literatura reportan validación cruzada dejando uno fuera, esta técnica de validación puede llevar a obtener conclusiones demasiado optimistas, por lo que se sugiere utilizar técnicas más adecuadas como es el caso de R_{10-cv}^2 (Golbraikh & Tropsha, 2002).

Tabla 5. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los mejores modelos construidos respecto a los modelos reportados en la literatura en el conjunto de datos de ecotoxicidad.

	pNOAEC*	Platyurus	48H ^e	96H ^f	120H ^g	132H ^h
<i>mejor QuBiLS-MAS^a</i>	0.86	<u>0.821</u>	0.85	<u>0.89</u>	<u>0.80</u>	0.70
<i>menor QuBiLS-MAS^a</i>	0.78	0.800	0.81	0.87	0.78	0.65
<i>promedio QuBiLS-MAS^a</i>	0.80	0.81	0.82	0.88	0.78	0.67
	(.030)**	(.008)	(.005)	(.004)	(.003)	(.012)
<i>Modelo 1 Nath et al. (Sección 3.1 en (Nath et al., 2022))^b</i>	0.66					
<i>PLS Lavado et al. (Tabla 1 en (Lavado et al., 2021))^c</i>		0.72				
<i>Liu et al. (Tabla 2 en (T. Liu et al., 2020))^d</i>			<u>0.84</u>	0.91	0.83	0.80

^a: validación cruzada con 10 pliegues (R_{10-cv}^2)

^b: validación dejando uno fuera (R_{10o}^2)

^c: validación cruzada con 5 pliegues (R_{5-cv}^2)

^d: Coeficiente de determinación (R^2)

^e: pLC50-48h-zebrafish

^f: pLC50-96h-zebrafish

^g: pLC50-120h-zebrafish

^h: pLC50-132-zebrafish

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

Adicionalmente se puede observar que los valores de R_{10-cv}^2 obtenidos por los modelos para los conjuntos de datos pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish y pLC50-132-zebrafish fue menor que el rendimiento de los modelos reportados (T. Liu et al., 2020). Sin embargo, el rendimiento de los modelos reportados corresponde al coeficiente de determinación (R^2) sin validación

crusada y, por lo tanto, el rendimiento reportado puede ser irreal (Tropsha et al., 2003). Aun así, los valores de R_{10-cv}^2 obtenidos por los modelos creados para predecir pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish fueron a lo sumo 3,43%, 3,98% y 5,84% menor que el respectivo valor R^2 reportado. En el caso de pLC50-132h-zebrafish, los modelos construidos fueron al menos un 13,34% inferiores al valor de R^2 reportado. Pero como se mencionó anteriormente, R^2 no es una métrica adecuada para evaluar la calidad de los modelos predictivos. De hecho, se puede observar que el valor R_{10-cv}^2 reportado sobre el conjunto completo (entrenamiento + prueba, 143 compuestos) por los autores de este modelo fue igual a 0.678 (ver Tabla 2 en (T. Liu et al., 2020)), que es comparable a los valores R_{10-cv}^2 alcanzados por nuestros modelos en los 114 compuestos de entrenamiento.

Además, las Tablas 6 y 7 muestran el rendimiento (Q_{test}^2) del mejor y peor modelo, así como el rendimiento promedio de los mejores modelos para cada uno de los conjuntos de prueba. Además, se muestra el rendimiento de los mejores modelos reportados en la literatura en estos conjuntos de prueba (Ansari & Palmer, 2018; Bonachéra & Horvath, 2008; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; Gupta et al., 2016; Klamt et al., 2012; Lavado et al., 2021; T. Liu et al., 2020; Martínez-Santiago et al., 2017; Nath et al., 2022; Sutherland et al., 2004; Tosco & Balle, 2012). Si se analiza el rendimiento de los modelos podemos observar como el mejor modelo tiene rendimiento superior a los modelos reportados en la literatura en doce de los catorce conjuntos de pruebas. En detalle, para los conjuntos de datos ACE, ACHE, BZR, COX2, DHFR, THER, THR, pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish, pNOAEC y pLC50-Platyurus el mejor modelo no cooperativo fue superior en 1,28 %, 12,5 %, 8,2 %, 12 %, 10,14 %, 8,22 %, 8,97%, 8%, 4%, 7%, 18% y 23%, respectivamente, que los mejores modelos reportados en la literatura.

También se puede observar que el rendimiento promedio (y el más bajo) de los modelos fueron mejores o similares que el rendimiento de los mejores modelos reportados en la literatura en un 6,94 % (1,39 %), 3,28 % (1,64 %), 8 % (2 %), 10,14 % (10,14 %), 6,85 % (5,48 %), 6,41 % (5,13 %), 5 %(similar), 2 %(similar), 5 %(1 %), 13 %(2 %)y 19 %(16 %), en los conjuntos de datos ACHE, BZR, COX2, DHFR, THER, THR, pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish, pNOAEC y pLC50-Platyurus, respectivamente. Para el conjunto de moléculas ACE, el rendimiento promedio (y el más bajo) de los modelos fue ligeramente inferior al del mejor modelo reportado en la literatura, pero superior al resto de los modelos reportados. Mientras que para GPB y pLC50-132h-zebrafish ningún modelo tuvo rendimiento superior respecto al mejor modelo reportado en la literatura, pero fueron superiores al resto de los modelos.

Tabla 6. Habilidades de generalización (Q_{test}^2) obtenidos por los mejores modelos construidos respecto los mejores modelos reportados en la literatura para el conjunto de datos de Sutherland

	ACE	ACHE	BZR	COX2	DHFR	GPB	THER	THR
<i>#mejor QuBiLS-MAS</i>	<u>0.79</u>	<u>0.81</u>	<u>0.66</u>	<u>0.56</u>	<u>0.76</u>	<u>0.77</u>	<u>0.79</u>	<u>0.85</u>
<i>##menor QuBiLS-MAS</i>	0.77	0.73	0.62	0.51	0.76	0.76	0.77	0.82
<i>###promedio QuBiLS-MAS</i>	0.77	0.77	0.63	0.54	0.76	0.76	0.78	0.83
	(.004)	(.022)	(.010)	(.011)	(.002)	(.004)	(.004)	(.006)
<i>QuBiLS-MIDAS (SI11 en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020))</i>	<u>0.78</u>	0.67	<u>0.61</u>	<u>0.50</u>	0.65	<u>0.83</u>	<u>0.73</u>	<u>0.78</u>
<i>GDI_s (Tabla 3 en (García-Jacas et al., 2019))</i>	0.70	<u>0.72</u>	0.39	0.44	<u>0.69</u>	0.68	0.66	0.74
<i>CoMFA (Tabla 4 en (Sutherland et al., 2004))</i>	0.49	0.47	0.00	0.29	0.59	0.42	0.54	0.63
<i>COMSIA basic (Tabla 4 en (Sutherland et al., 2004))</i>	0.52	0.44	0.08	0.03	0.52	0.46	0.36	0.55
<i>COMSIA extra (Tabla 4 en (Sutherland et al., 2004))</i>	0.49	0.44	0.12	0.37	0.53	0.59	0.53	0.63
<i>EVA (Tabla 4 en (Sutherland et al., 2004))</i>	0.36	0.28	0.16	0.17	0.57	0.49	0.36	0.11
<i>HQSAR (Tabla 4 en (Sutherland et al., 2004))</i>	0.30	0.37	0.17	0.27	0.63	0.58	0.53	-0.25
<i>2D (Tabla 4 en (Sutherland et al., 2004))</i>	0.47	0.16	0.14	0.25	0.47	-0.06	0.14	0.04
<i>2.5D (Tabla 4 en (Sutherland et al., 2004))</i>	0.51	0.16	0.20	0.27	0.49	0.04	0.07	0.28
<i>O3Q (Tabla 1 en (Tosco & Balle, 2012))</i>	0.69	0.67	0.17	0.32	0.60	0.50	0.51	0.67
<i>O3QMFA (Tabla 1 en (Klamt et al., 2012))</i>	0.45	0.61	0.13	0.37	0.59	0.29	0.49	0.60
<i>O3A/O3Q (Tabla 1 en (Tosco & Balle, 2012))</i>	0.54	0.65	0.24	0.28	0.53	0.41	-0.18	0.30
<i>COSMOsar3D (Tabla 1 en (Klamt et al., 2012))</i>	0.62	0.61	0.13	0.43	0.58	0.63	0.59	0.66
<i>2D-FPT (Tabla 3 en (Bonachéra & Horvath, 2008))</i>	0.71	0.71	0.38	0.33	0.68	0.67	0.65	0.74
<i>CARMa PLS (Tabla 4 en (Ansari & Palmer, 2018))</i>	0.64	0.67	0.21	0.38	0.56	-	-	-
<i>CARMa GA-PLS (Tabla 4 en (Ansari & Palmer, 2018))</i>	0.62	0.70	0.21	0.35	0.57	-	-	-
<i>CARMa RF (Tabla 4 en (Ansari & Palmer, 2018))</i>	0.64	0.54	0.22	0.38	0.65	-	-	-
<i>MACCS-ANN-QSAR (Tabla 12 en (Gupta et al., 2016))</i>	0.08	0.04	0.06	0.23	0.48	-	-	-

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

Tabla 7. Habilidades de generalización (Q_{test}^2) obtenidos por los mejores modelos construidos y los mejores modelos reportados en la literatura para los conjuntos de compuestos ecotoxicológicos

	pNOAEC*	Platyurus	48H	96H	120H	132H
#mejor QuBiLS-MAS	0.86	0.94	0.95	0.96	0.93	0.78
##menor QuBiLS-MAS	0.70	0.89	0.87	0.92	0.87	0.70
###promedio QuBiLS-MAS	0.81 (.065)*	0.92 (.014)	0.93 (.007)	0.95 (.010)	0.91 (.016)	0.73 (.023)
Modelo 1 Nath et al. (Sección 3.1 en (Nath et al., 2022))	0.68					
PLS Lavado et al. (Tabla 1 en (Lavado et al., 2021))		0.74				
Liu et al. (Tabla 2 en (T. Liu et al., 2020))			0.87	0.92	0.85	0.81

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

3.4.4 Estudio estadístico basado en estimación bayesiana

Tradicionalmente se ha supuesto que las pruebas estándar de hipótesis nula y las pruebas post-hoc determinan si existe una diferencia estadísticamente significativa entre diferentes enfoques. Estas pruebas presentan diferentes desventajas asociadas a la cantidad de datos utilizadas o al valor de *p-value* considerado (Benavoli et al., 2017). Debido a esto se propuso recientemente el uso de técnicas bayesianas para realizar estos análisis estadísticos (Benavoli et al., 2017). Un análisis por pares fue realizado para este estudio, donde se calcula la probabilidad de que un enfoque supere a otro según los resultados obtenidos. Consulte la sección 2 en (Benavoli et al., 2017) para una discusión en profundidad. La distribución de probabilidad obtenida a través del análisis bayesiano se muestrea vía Monte Carlo y se muestra en coordenadas baricéntricas, donde se distinguen tres áreas: una donde el desempeño de un enfoque es mejor que el otro considerado, o viceversa (segunda área); y una tercera región donde los dos enfoques se consideran estadísticamente equivalentes. Esta región de equivalencia se configura a través del parámetro *rope*. Este parámetro establece la diferencia mínima necesaria entre los valores de rendimiento de los enfoques comparados para que se consideren significativamente diferentes entre sí. En este trabajo se establece un valor de *rope* igual a 0.01.

El rendimiento en los conjuntos de prueba (Q_{test}^2) de los modelos construidos se comparó con el rendimiento de los mejores modelos reportados en la literatura respecto a los catorce conjuntos de datos. Para esto se tuvieron en cuenta los valores de Q_{test}^2 más altos, promedio y más bajos obtenidos por los modelos construidos (ver Tabla 8). En general se puede concluir que los modelos creados a partir de conjuntos de DMs obtenidos a través del algoritmo AExOp-DCS presentan una mayor probabilidad de lograr mejores desempeños que los modelos creados a partir de conjuntos de DMs obtenidos sin optimizar los DCS (modelos del estado del arte).

Específicamente, se puede observar cómo la probabilidad de que el mejor modelo tenga rendimiento superior es del 100 %. De igual forma, si el rendimiento en promedio de los modelos propuestos es analizado, se puede observar que hay un 98 % de probabilidad de que en promedio los modelos construidos a partir de un conjunto de DMs optimizados sea superior al rendimiento de los modelos construidos a partir de DMs sin optimizar. Además, se puede observar que incluso los modelos construidos con peor desempeño son superiores a los mejores modelos del estado del arte con una probabilidad superior a 0,79.

Tabla 8. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos no cooperativos respecto a los mejores modelos reportados en la literatura.

<i>QuBiLS-MAS</i>	Literatura	$p(\text{QuBiLS} - \text{MAS})^a$	$p(\text{Rope})^b$	$p(\text{literatura})^c$
Mejor	Mejor	1.00	0.00	0.00
Promedio	Mejor	0.98	0.00	0.02
Menor	Mejor	0.79	0.07	0.14

^a: probabilidad de que el modelo basado en el conjunto optimizado sea significativamente mejor

^b: probabilidad de que ambos seas similares

^c: probabilidad de que el mejor modelo de la literatura sea superior

Por lo tanto, los resultados estadísticos indican que la estrategia AExOp-DCS es una mejor alternativa que el enfoque aplicado actualmente (es decir, sobrecálculo de DMs y luego selección) para determinar buenos conjuntos de DMs para el modelado QSAR. Esto se debe a que se pueden construir modelos con rendimientos notablemente superiores a partir de los conjuntos "óptimos" de DMs obtenidos a través del algoritmo C-AExOp-DCS. Es necesario considerar que la superioridad estadística del rendimiento de los modelos construidos se obtuvo utilizando solo MLR, que es más simple que varias de las técnicas estadísticas o de aprendizaje (p. ej., PLS, GBM, RF) utilizadas para construir los modelos reportados en la literatura. En consecuencia, el rendimiento superior obtenido se debió principalmente a la calidad de los conjuntos "óptimos" de DMs obtenidos a través de las estrategias propuestas.

3.4.5 Análisis de los mejores modelos por cada conjunto de datos

La **Tabla 9** muestra el rendimiento de los mejores modelos en cada conjunto de datos químicos estudiados. El coeficiente de determinación (R^2), el coeficiente de determinación ajustado (R_{adj}^2), el coeficiente de determinación en validación cruzada con 10 pliegues (R_{10-cv}^2), el error absoluto medio con validación cruzada (MAE_{10-cv}), el error cuadrático medio con validación cruzada ($RMSE_{10-cv}$), y el coeficiente de determinación con validación cruzada con valores de actividad perturbados ($R_{y-scramb}^2$) son las métricas de rendimiento utilizadas para determinar la calidad de los modelos en el conjunto de datos de entrenamiento. Las técnicas de validación cruzada en 10 pliegues (J. H. Kim, 2009) y *Y-scrambling* (Rücker et al., 2007) se repitieron cada una 100 veces para determinar los valores de R_{10-cv}^2 , MAE_{10-cv} , $RMSE_{10-cv}$ y $R_{y-scramb}^2$, respectivamente. Las métricas Q_{test}^2 , MAE_{test} y $RMSE_{test}$ son las calculadas para evaluar la capacidad de generalización de los modelos. Es importante señalar, que dado la calidad de los modelos que se van a analizar en esta sección, estos van a estar disponibles en las herramientas computacionales SiLiS-PREENZA (Yovani Marrero Ponce, 2023a) y SiLiS-PTOXRA (Yovani Marrero Ponce, 2023b), respectivamente.

Como resultado, se puede observar en la **Tabla 9** que todos los modelos obtuvieron buenas métricas de desempeño en el conjunto de entrenamiento, presentando en su mayoría valores de R^2 , R_{adj}^2 y R_{10-cv}^2 entre 0.7 y 0.9. Solo los modelos para predecir pIC50-BZR, pIC50-COX2 y pLC50-132h-zebrafish obtuvieron valores R_{10-cv}^2 ligeramente inferiores a 0,7. Nótese además que los valores de Q_{test}^2 indican que todos los modelos lograron buenas habilidades de generalización.

Específicamente, exceptuando los modelos para predecir pIC50-BZR ($Q_{test}^2 = 0,6597$) y pIC50-COX2 ($Q_{test}^2 = 0,5641$), los valores de Q_{test}^2 obtenidos por los demás modelos oscilan entre 0,76 y 0,95. Es importante señalar que los modelos para predecir pIC50-BZR y pIC50-COX2 presentan valores de Q_{test}^2 superiores a los mejores modelos propuestos hasta la fecha. En general, el rendimiento de estos modelos sugiere un buen balance bias-varianza ya que no existe diferencia entre los valores de R_{10-cv}^2 y Q_{test}^2 . Además, se puede observar que los valores de $R_{y-scramb}^2$ siempre fueron inferiores a 0,05, lo que indica que los DMs utilizados en estos modelos no se correlacionan aleatoriamente con las actividades modeladas y, por lo tanto, el buen desempeño de ellos no es casuístico.

Tabla 9. Valores de rendimiento en entrenamiento, validación y DA alcanzados por los mejores modelos en cada conjunto de datos.

Conjunto de datos	Conjunto de entrenamiento					Conjunto de prueba ^c		
	R^2	R^2_{10-cv} ^a	MAE_{10-cv} ^a	$RMSE_{10-cv}$ ^a	$R^2_{y-scramb}$ ^b	Q^2_{test}	MAE_{test}	$RMSE_{test}$
pIC ₅₀ -ACE	0.855	0.8200 (.005)	0.7614 (.013)	0.9955 (.015)	0.0316 (.038)	0.7879 0.7322 (4)	0.8914 0.9079	1.1747 1.1936
pIC ₅₀ -ACHE	0.808	0.7378 (.020)	0.5146 (.014)	0.6223 (.026)	0.0345 (.046)	0.8115 0.7955 (4)	0.4554 0.4258	0.6004 0.5723
pIC ₅₀ -BZR	0.770	0.6915 (.013)	0.2952 (.006)	0.3733 (.010)	0.0191 (.028)	0.6597 0.6493 (3)	0.4679 0.4625	0.6353 0.6341
pIC ₅₀ -COX2	0.725	0.6910 (.006)	0.4473 (.004)	0.5659 (.005)	0.0118 (.020)	0.5641 0.5841 (6)	0.7986 0.7929	1.0450 1.0451
pIC ₅₀ -DHFR	0.788	0.7648 (.003)	0.4506 (.004)	0.6142 (.004)	0.0125 (.014)	0.7641 0.7733 (3)	0.5288 0.5226	0.6621 0.6552
pK _i -GPB	0.872	0.8052 (.013)	0.3872 (.015)	0.4765 (.017)	0.0369 (.048)	0.7700 0.7620 (4)	0.4610 0.5074	0.6010 0.6518
pK _i -THERM	0.817	0.7458 (.011)	0.7289 (.021)	0.9513 (.022)	0.0466 (.062)	0.7924 0.7058 (4)	0.7387 0.7731	1.0242 1.0789
pK _i -THR	0.897	0.8483 (.007)	0.3114 (.008)	0.3736 (.010)	0.0360 (.051)	0.8494 0.8098 (3)	0.3123 0.3414	0.4351 0.4585
pLC ₅₀ -48h-zebrafish	0.860	0.8036 (.014)	0.4931 (.015)	0.6695 (.027)	0.0104 (.014)	0.9308 0.9175 (2)	0.3003 0.2994	0.4306 0.4338
pLC ₅₀ -96h-zebrafish	0.914	0.8484 (.039)	0.4654 (.030)	0.6267 (.069)	0.0313 (.035)	0.9628 0.9644 (1)	0.6125 0.6408	0.7330 0.7576
pLC ₅₀ -120h-zebrafish	0.839	0.7389 (.025)	0.4930 (.021)	0.6630 (.040)	0.0141 (.017)	0.9264 0.9243 (2)	0.3711 0.3545	0.5374 0.5137
pLC ₅₀ -132h-zebrafish	0.777	0.6610 (.021)	0.3727 (.012)	0.4907 (.019)	0.0157 (.021)	0.7801 0.7941 (4)	0.4552 0.4204	0.5697 0.5379
pNOAEC	0.816	0.7679 (.008)	0.5207 (.011)	0.6586 (.013)	0.0270 (.035)	0.7869 0.7758 (1)	0.3982 0.3466	0.5448 0.4314
pLC ₅₀ -T.platyurus	0.837	0.7868 (.011)	0.3618 (.014)	0.4634 (.015)	0.0323 (.046)	0.9446	0.3799	0.4477

^a: promedio luego de ejecutar 100 veces validación cruzada con 10 pliegues. La desviación estándar se muestra entre paréntesis.

^b: R^2_{10-cv} promedio de 100 ejecuciones del modelo original después de perturbar los valores de la actividad. La desviación estándar se muestra entre paréntesis.

^c: La segunda fila por modelo es el rendimiento del modelo considerando los compuestos dentro del dominio de aplicabilidad. Entre paréntesis se muestra el número de compuestos que no caen dentro. Todos los compuestos en el conjunto de pruebas de pLC₅₀-T.platyurus están dentro del DA.

3.5 Conclusiones parciales

Con el objetivo de determinar un subconjunto “óptimo” de DMs se presentó en este capítulo el algoritmo AExOp-DCS. Este algoritmo optimiza el DCS de los DMs analizados mediante la exploración de los diferentes valores que pueden ser tomados por cada uno de los parámetros de los cuales dependen los algoritmos de cálculo de descriptores.

Además, se examinó la calidad de los subconjuntos de DMs “optimizados” mediante AExOp-DCS en la construcción de modelos para predecir diferentes actividades biológicas en catorce conjuntos de compuestos químicos. Para esto, una vez ejecutado el algoritmo para cada uno de los conjuntos de compuestos y determinado los respectivos subconjuntos de DMs optimizados, se procedió a la construcción de diferentes modelos QSAR utilizando Regresión lineal múltiple como técnica estadística. Los resultados fueron comparados con los modelos disponibles en la literatura tomando en cuenta el rendimiento observado tanto en el conjunto de entrenamiento como en los conjuntos de prueba.

Por lo general, se observó que los rendimientos de los modelos construidos tuvieron mejor desempeño tanto en bondad de ajuste como en habilidades de generalización que los modelos reportados. Adicionalmente, se validaron estadísticamente los valores de Q_{ext}^2 de los modelos construidos respecto a los mejores modelos del estado del arte. De esta manera se pudo observar cómo los modelos construidos a partir de los DMs devueltos tienen mayor probabilidad de tener rendimientos superiores que los mejores modelos del estado del arte.

Capítulo 4. Un enfoque coevolutivo para determinar un conjunto óptimo de descriptores moleculares mediante la exploración y optimización del Espacio de Configuración del Descriptor

El algoritmo AExOp-DCS optimiza diferentes DCSs, los cuales están representando mediante una población de cromosomas. Estos DCSs son teóricamente diferentes, por lo que no es posible recombinar cromosomas que pertenezcan a diferentes poblaciones. Esto implica que cada DCS se analiza de forma independiente, lo cual es irreal en el contexto del modelado QSAR. En primer lugar, porque es bien sabido que no existe una sola familia (algoritmo) de DM capaz de codificar toda la información química para diferentes conjuntos de compuestos; y, en segundo lugar, porque la predicción de nuevos compuestos no se puede expresar como la suma de los aportes individuales de los DMs, sino en el aporte colectivo de estos. Lo antes mencionado constituye precisamente un aspecto a mejorar en el algoritmo *AExOp-DCS* ya que no considera la sinergia existente entre DMs teóricamente diferentes durante el proceso evolutivo. En este capítulo se presenta el algoritmo C-AExOp-DCS, el cual optimiza el DCS a partir de conceptos relacionados a las estrategias cooperativas (Potter & Jong, 1994; Rodríguez-Coayahuitl et al., 2020; Wiegand & Jong, 2004). De igual forma, se realiza un análisis de la calidad de los subconjuntos devueltos por dicha estrategia considerando el rendimiento de los modelos que se construyen a partir de los subconjuntos retornados.

4.1 C-AExOp-DCS: Algoritmo Coevolutivo para Explorar y Optimizar el Espacio de configuración del Descriptor

En este capítulo se propone un algoritmo basado en AExOp-DCS denominado *C-AExOp-DCS*. Dicho algoritmo añade conceptos relacionados con los algoritmos cooperativos-coevolutivos (ACC) (Potter & Jong, 2000, 1994; Rodríguez-Coayahuitl et al., 2020; Wiegand & Jong, 2004) a AExOp-DCS con el objetivo de mejorar el proceso de optimización (ver Algoritmo 5). Esto se realiza teniendo en cuenta que en los enfoques cooperativos la importancia de cada individuo se basa en su relación (interacción) con individuos de otras poblaciones (Potter & Jong, 1994; Wiegand & Jong, 2004). Por lo que la relevancia (aptitud) de cada DM (cromosoma) se va a determinar ahora teniendo en cuenta también su interacción con descriptores de otros DCS. Nuestra hipótesis es que los modelos que se construyan a partir de los

subconjuntos devueltos por la estrategia cooperativa tendrán mayor poder predictivo que los modelos que se construyan a partir del enfoque no cooperativo.

Algoritmo 5. Seudo Código del algoritmo C-AExOp-DCS.

Entrada

- *Dataset* ← conjunto de compuestos de entrenamiento
- *Activity* ← actividad/propiedad biológica de los compuestos
- *DCSs* ← lista de los DCSs

Salida

- *Subconjunto best_subset* con los mejores DMs encontrados

1. $t \leftarrow 0$
2. $best_fitness \leftarrow 0$
3. $best_solution \leftarrow empty$
4. $populations \leftarrow size(DCSs)$
5. *for* $i \leftarrow 1$ to $size(DCSs)$
6. $populations_{i,0} \leftarrow initialize(DCSs_i)$ /* inicializar de forma aleatoria /
7. *end for*
8. *while* $t \leq T$
9. *for* $i \leftarrow 1$ to $size(populations)$
10. $phenotypes \leftarrow compute_descriptors(populations_{i,t}, Dataset)$
11. /* start share MD process between populations */
12. $phenotypes_to_send \leftarrow get_phenotypes_for_coop(population_i)$
13. $phenotypes_coop = []$
14. *for* $j \leftarrow 1$ to $size(populations), j! = i$
15. $send_phenotypes_for_coop(phenotypes_to_send, population_j)$
16. $phenotypes_coop.add($
17. $recv_phenotypes_for_coop(phenotypes_to_send, population_j))$
18. *end for*
19. /* end share MD process between populations */
20. $fitness_values \leftarrow compute_chromosome_fitness(phenotypes, phenotypes_coop, Activity)$
21. *end for*
22. $merged_pool \leftarrow join(phenotypes, best_solution)$
23. $temp_best_solution, temp_best_fitness \leftarrow CFS_method(merged_pool, Activity)$
24. *if* $temp_best_fitness > best_fitness$
25. $best_fitness \leftarrow temp_best_fitness$
26. $best_solution \leftarrow temp_best_solution$
27. *end if*
28. *if* $reset_populations$
29. *for* $i \leftarrow 1$ to $size(DCSs)$
30. $populations_{i,t+1} \leftarrow initialize(DCSs_i)$
31. *end for*
32. *else*
33. *for* $i \leftarrow 1$ to $size(DCSs)$
34. $parents \leftarrow tournament_selection(populations_{i,t}, fitness_values)$
35. $offspring \leftarrow HUX_crossover(parents)$
36. $offspring' \leftarrow mutation(offspring)$
37. $populations_{i,t+1} \leftarrow replacement(populations_{i,t}, parents, offspring')$
38. *end for*
39. *end if*
40. $t \leftarrow t + 1$
41. *end while*
42. *return* $best_solution$

El proceso cooperativo implica que determinados cromosomas de una población sean enviados al resto de las poblaciones. La cantidad de cromosomas que se compartirán es un valor definido por el

usuario, y se seleccionan a través del mismo método de selección que se utiliza en el proceso evolutivo. Por lo tanto, cada población recibirá un total de $s * (p - 1)$ DMs, donde p es el número de poblaciones y s es el número de DMs seleccionados de cada población. De esta forma, el valor de calidad de cada DM se calculará respecto tanto a los DMs de su misma población como a los DMs de poblaciones diferentes. Para determinar el valor de calidad de los DMs se consideraron cuatro criterios para luego integrarlos utilizando la integral de Choquet (Choquet, 1954; Grabisch & Labreuche, 2010) (ver sección 3.1).

Un algoritmo coevolutivo debe satisfacer dos condiciones. El primero establece que el número mínimo de poblaciones es dos (Wiegand & Jong, 2004), mientras que la segunda establece que la función de aptitud para un individuo (cromosoma) debe depender de su relación con otros individuos (medida de función subjetiva) (Wiegand & Jong, 2004). En este sentido, tres de las cuatro medidas de calidad no cumplen esta segunda condición: *ReliefF*, SE y correlación de Pearson, mientras que MDI sí. MDI calcula la reducción total de pérdida o impureza de una variable en particular cada vez que se utiliza para crear un nodo en la construcción de un modelo *Random Forest* (Wiegand & Jong, 2004). En este sentido, se ha demostrado que la importancia MDI de una variable es invariante con respecto a agregar o eliminar variables irrelevantes, pero no cuando se incluyen o eliminan variables relevantes (Wiegand & Jong, 2004). Por lo tanto, si cromosomas relevantes son compartidos entre las distintas poblaciones se espera que esto influya en el valor de calidad de los DMs.

La Figura 6 muestra cómo cambian los valores de MDI cuando se incluyen nuevos DMs en el análisis, y cómo estos cambios afectan la agregación basada en la integral de Choquet, garantizando la condición de medida subjetiva necesaria para estar en presencia de un enfoque cooperativo (Wiegand & Jong, 2004). Por un lado, la Figura 6A muestra una población de tres DMs y sus valores de importancia calculados con la integral de Choquet. La Figura 6A también muestra el valor de los criterios MDI, Relief-F, SE y correlación de Pearson para cada DM. Aquí podemos ver que los valores de MDI para cada DM son 0,49, 0,51 y 0, mientras que los valores de Choquet son 0,63, 0,64 y 0,43. Por otro lado, la Figura 6B muestra cómo se calcula la relevancia de un DM en un entorno cooperativo. Como puede verse, primero es necesario compartir cromosomas entre poblaciones para evaluar la calidad de un DM con respecto a los DMs de otras poblaciones. La Figura 6B muestra cómo se envían dos DMs, D_{23} de una población y D_{32} de la otra población para evaluar la calidad de los DMs D_{11} , D_{12} y D_{13} en un entorno cooperativo. En la Figura 6B es posible ver cómo cambian los valores de MDI para D_{11} , D_{12} y D_{13} y cómo estos cambios afectan los valores de Choquet. Por lo tanto, debido a que una de las cuatro medidas de calidad cumple con la condición de medida de la función subjetiva, se puede concluir que nuestro enfoque de agregación basado en la integral

de Choquet para calcular el valor de aptitud de cada cromosoma también la cumple. En consecuencia, el algoritmo *C-AExOp-DCS* es un enfoque de coevolución cooperativa.

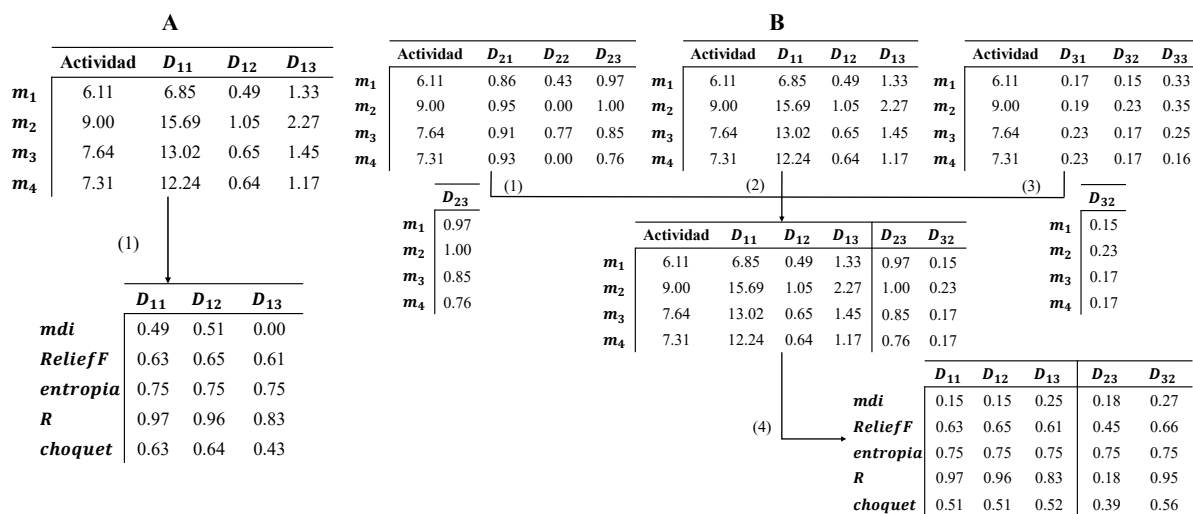


Figura 6. Ejemplo de (A) cálculo de la calidad del DM sin considerar DMs de otras poblaciones. (B) Cálculo de la calidad de un DM considerando DMs de otras poblaciones.

4.2 Validación del algoritmo C-AExOp-DCS en la construcción de modelos

QSAR.

4.2.1 Procedimiento para validar la calidad de los descriptores moleculares retornados

Para garantizar la comparabilidad de los resultados con respecto a los modelos ya reportados en la literatura y los modelos reportados en el capítulo anterior se recrearon los mismos escenarios en los que estos fueron construidos. El algoritmo de búsqueda cooperativo fue ejecutado utilizando la misma configuración utilizada para la ejecución del algoritmo AExOp-DCS definido en la Tabla 3. Los modelos fueron construidos a partir de la técnica estadística de regresión lineal múltiple (MLR) utilizando la misma cantidad de descriptores moleculares utilizados en la construcción de los modelos no cooperativos (modelos reportados en el Capítulo 3). En el caso de los conjuntos de compuestos de Sutherland se escogieron los mejores 30 mejores modelos. Para los conjuntos de compuestos para la predicción de

ecotoxicidad se escogieron aquellos modelos con rendimiento similar o superior a los modelos no cooperativos.

El rendimiento de los modelos cooperativos fue comparado respecto a los modelos no cooperativos reportados en el Capítulo 3 y a los mejores modelos del estado del arte. Para esto se analizó el rendimiento del mejor modelo, el peor modelo y el rendimiento en promedio de los modelos obtenidos por cada conjunto de datos tanto en el conjunto de entrenamiento como en el de prueba. Para garantizar un buen balance bias-varianza se tomaron los modelos cuya diferencia entre el rendimiento en los conjuntos de entrenamiento y prueba no fuera mayor a 0.05. El rendimiento de los modelos en conjunto de entrenamiento y prueba fue medido utilizando el coeficiente de determinación con validación cruzada a partir de 10 pliegues (R_{10-cv}^2) y validación externa (Q_{ext}^2), respectivamente.

4.2.2 Evaluación de los modelos construidos y análisis respecto a modelos definidos en la literatura.

Primero se utilizaron los conjuntos de entrenamiento para comparar el rendimiento de los modelos cooperativos respecto a los modelos reportados en el Capítulo 3 y los modelos del estado del arte. La Tabla 10 muestra el rendimiento en el conjunto de entrenamiento (R_{10-cv}^2) del mejor modelo, el peor modelo y el rendimiento promedio de los mejores modelos cooperativos. Estos se compararon con respecto a los modelos del estado del arte y con respecto a los modelos no cooperativos utilizando los conjuntos de Sutherland (Sutherland et al., 2004). La Tabla 11 muestra el rendimiento en el conjunto de entrenamiento del mejor modelo, el peor modelo y el rendimiento en promedio de los modelos cooperativos y los compara respecto a los modelos del estado del arte y a los modelos no cooperativos para los conjuntos de compuestos ecotoxicos (Lavado et al., 2021; T. Liu et al., 2020; Nath et al., 2022).

Si se compara el rendimiento de los modelos cooperativos respecto a los modelos del estado del arte se puede observar como todos los modelos cooperativos obtuvieron mejor rendimiento en los conjuntos de entrenamiento ACHE, BZR, DHFR, THER, pLC50-Platyurus y pLC50-48h-zebrafish. Adicionalmente se puede notar que el mejor modelo cooperativo es superior a los modelos del estado del arte para los conjuntos de entrenamiento ACE y COX2. En el caso de los conjuntos GPB, THR, pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish y pLC50-132h-zebrafish, los valores R_{10-cv}^2 alcanzados por los modelos cooperativos son inferiores a los valores reportados en la literatura. Pero como se puede observar, los valores reportados en la literatura corresponden a la métrica R_{LOO}^2 en el caso de

GPB y THR, y al coeficiente de determinación (R_2) en el caso del pLC50-48h-zebrafish, pLC50-96h-zebrafish, pLC50-120h-zebrafish y pLC50-132h-zebrafish. Como se ha mencionado anteriormente estas métricas pueden llevar a obtener conclusiones sobreoptimistas (Golbraikh & Tropsha, 2002).

Tabla 10. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos cooperativos y no cooperativos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de Sutherland.

	ACE	ACHE	BZR	COX2	DHFR	GPB	THER	THR
&Mayor QuBiLS-MAS (Cooperativo) ^a	<u>0.82</u> *	0.82	<u>0.72</u>	<u>0.67</u>	0.74	<u>0.82</u>	0.81	0.84
&&Menor QuBiLS-MAS (Cooperativo) ^a	0.8	0.78	0.71	0.66	0.72	0.8	0.77	0.83
&&&promedio QuBiLS-MAS (Cooperativo) ^a	0.81 (.006)**	0.79 (.007)	0.71 (.002)	0.67 (.004)	0.72 (.004)	0.81 (.006)	0.79 (.008)	0.83 (.003)
#mayor QuBiLS-MAS (No cooperativo) ^a	0.83	<u>0.78</u>	0.73	0.7	0.77	0.83	<u>0.76</u>	<u>0.87</u>
##menor QuBiLS-MAS (No cooperativo) ^a	0.82	0.73	0.69	0.69	0.76	0.79	0.74	0.83
###promedio QuBiLS-MAS (No cooperativo) ^a	0.83 (.004)	0.74 (.012)	0.70 (.008)	0.69 (.003)	0.76 (.002)	0.81 (.010)	0.75 (.006)	0.85 (.008)
QuBiLS-MIDAS (SI11 en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020)) ^b	0.75	0.66	0.71	0.63	0.7	0.83	0.75	0.81
GDIs (Tabla 3 en (García-Jacas et al., 2019)) ^b	<u>0.82</u>	<u>0.78</u>	0.7	<u>0.67</u>	0.72	0.83	0.74	0.89
jCompoundMapper (Tabla 6 en (Hinselmann et al., 2011)) ^b	0.69	0.57	0.56	0.55	<u>0.76</u>	0.53	0.53	0.58

^a: validación cruzada con 10 pliegues (R_{10-cv}^2)

^b: validación dejando uno fuera (R_{100}^2)

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos no cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

& : Mejor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&& : Peor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&&& : Promedio del rendimiento de los mejores modelos cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por C-AExOp-DCS

De igual forma se puede realizar una comparación entre los modelos cooperativos y no cooperativos respecto al rendimiento de estos sobre los conjuntos de entrenamiento (R_{10-cv}^2). En las Tablas 10 y 11 se puede observar cómo los modelos no cooperativos tienen rendimiento superior en los conjuntos ACE, COX2, DHFR y pLC50-96h-zebrafish, mientras que los modelos cooperativos son superiores en los conjuntos de entrenamiento ACHE, THER y pLC50-132h-zebrafish. Si se analiza el rendimiento del mejor modelo cooperativo y no cooperativo se puede observar que para los conjuntos de datos ACHE, THER, pLC50-Platyurus, pLC50-120h-zebrafish y pLC50-132h-zebrafish el mejor modelo cooperativo es superior al mejor modelo no cooperativo en un 4%, 5%, 1%, 4% y 8%, respectivamente. Por el contrario, se puede ver cómo el mejor modelo no cooperativo tiene un rendimiento superior en un 1%, 1%, 3%, 3%, 1%, 3%,

4 %, 1 % y 2 % para los conjuntos de datos ACE, BZR, COX2, DHFR, GPB, THR, pNOAEC, pLC50-48h-zebrafish y pLC50-96h-zebrafish, respectivamente.

Tabla 11. Comparación del rendimiento en el conjunto de entrenamiento del mejor, peor y en promedio de los 30 mejores modelos cooperativos y no cooperativos respecto a los mejores modelos reportados en la literatura en el conjunto de datos de ecotoxicidad.

	pNOAEC	Platyurus	48H	96H	120H	132H
&Mayor QuBiLS-MAS (Cooperativo) ^a	0.82*	0.83	0.84	0.87	0.84	0.78
&&Menor QuBiLS-MAS (Cooperativo) ^a	0.78	0.80	0.81	0.84	0.78	0.70
&&&promedio QuBiLS-MAS (Cooperativo) ^a	0.79 (0.01) **	0.81 (0.02)	0.83 (.004)	0.85 (.005)	0.81 (.086)	0.74 (.009)
#mayor QuBiLS-MAS (No cooperativo) ^a	0.86	0.821	0.85	0.89	0.80	0.70
##menor QuBiLS-MAS (No cooperativo) ^a	0.78	0.800	0.81	0.87	0.78	0.65
###promedio QuBiLS-MAS(No cooperativo) ^a	0.80 (.0298)	0.81 (.008)	0.82 (.005)	0.88 (.004)	0.78 (.003)	0.67 (.012)
Modelo 1 Nath et al. (Sección 3.1 en (Nath et al., 2022)) ^b	0.66					
PLS Lavado et al. (Tabla 1 en (Lavado et al., 2021)) ^c		0.72				
Liu et al. (Tabla 2 en(T. Liu et al., 2020)) ^d			0.84	0.91	0.83	0.80

^a: validación cruzada con 10 pliegues (R_{10-cv}^2)

^b: validación dejando uno fuera (R_{100}^2)

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos no cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

& : Mejor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&& : Peor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&&& : Promedio del rendimiento de los mejores modelos cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por C-AExOp-DCS

Para evaluar la capacidad de generalización se compararon los modelos cooperativos respecto a los modelos del estado del arte y los modelos no cooperativos reportados en el Capítulo 3, teniendo en cuenta el rendimiento de estos en los conjuntos de prueba (Q_{test}^2). Para esto, se analizaron el rendimiento del mejor modelo, el peor modelo y el rendimiento promedio de los modelos cooperativos. Las Tablas 12 y 13 muestran el rendimiento (Q_{test}^2) del mejor y peor modelo no cooperativo y cooperativo, respectivamente, así como el rendimiento promedio de los mejores modelos para cada uno de los conjuntos de prueba. Además, se muestra el rendimiento de los mejores modelos reportados en la literatura en estos conjuntos de prueba (Ansari & Palmer, 2018; Bonachéra & Horvath, 2008; García-Jacas, Marrero-Ponce, Brizuela, et al., 2020; Gupta et al., 2016; Klamt et al., 2012; Lavado et al., 2021; T. Liu et al., 2020; Martínez-Santiago et al., 2017; Nath et al., 2022; Sutherland et al., 2004; Tosco & Balle, 2012).

Si se compara el rendimiento de los modelos cooperativos respecto a los modelos del estado del arte se puede observar como el mejor modelo cooperativo es superior para cada conjunto de prueba. Por otro lado, se puede observar que para los conjuntos de prueba de BZR, COX2, DHFR, THER, THR, pLC50-48h-zebrafish, pLC50-120h-zebrafish, pLC50-132h-zebrafish, pNOAEC y pLC50-Platyurus, el modelo cooperativo con menor valor de Q_{test}^2 es superior a los modelos del estado del arte.

Tabla 12. Habilidades de generalización (Q_{test}^2) obtenidos por los mejores modelos no cooperativos y cooperativos y los mejores modelos reportados en la literatura para el conjunto de datos de Sutherland

	ACE	ACHE	BZR	COX2	DHFR	GPB	THER	THR
&Mayor QuBiLS-MAS (Cooperativo)	0.79*	0.74	0.69	0.58	0.75	0.87	0.79	0.83
&&Menor QuBiLS-MAS (Cooperativo)	0.77	0.72	0.65	0.55	0.74	0.81	0.77	0.83
&&&promedio QuBiLS-MAS (Cooperativo)	<u>0.78</u> (0.04) **	0.72 (.004)	0.66 (008)	0.56 (.007)	0.75 (.002)	0.85 (.014)	0.78 (.006)	0.83 (.002)
#mayor QuBiLS-MAS (No cooperativo)	0.79	0.81	<u>0.66</u>	<u>0.56</u>	0.76	0.77	0.79	0.85
##menor QuBiLS-MAS (No cooperativo)	0.77	0.73	0.62	0.51	0.76	0.76	0.77	0.82
###promedio QuBiLS-MAS (No cooperativo)	0.77 (.004)	0.77 (.022)	0.63 (.010)	0.54 (.011)	0.76 (.002)	0.76 (.004)	0.78 (.004)	0.83 (.006)
QuBiLS-MIDAS (SI11 en (García-Jacas, Marrero-Ponce, Brizuela, et al., 2020))	<u>0.78</u>	0.67	0.61	0.50	0.65	<u>0.83</u>	<u>0.73</u>	0.78
GDIs (Tabla 3 en (García-Jacas et al., 2019))	0.70	0.72	0.39	0.44	0.69	0.68	0.66	0.74

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

: Mejor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Peor modelo no cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

: Promedio del rendimiento de los mejores modelos no cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

& : Mejor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&& : Peor modelo cooperativo construido a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por AExOp-DCS

&&& : Promedio del rendimiento de los mejores modelos cooperativos construidos a partir del subconjunto óptimo de DMs QuBiLS-MAS retornado por C-AExOp-DCS

Si son comparadas las capacidades de generalización de los modelos no cooperativos y cooperativos, se puede observar cómo los modelos cooperativos presentan rendimiento comparable o superior a los modelos no cooperativos. En específico, si es analizado el mejor modelo por cada estrategia, se puede observar cómo los modelos cooperativos son mejores en los conjuntos de pruebas BZR, COX2, GPB, pNOAEC, pLC50-Platyurus, pLC50-48h-zebrafish, pLC50-120h-zebrafish y pLC50-132h-zebrafish por un 3%, 2%, 10%, 3%, 2%, 1%, 3% y un 12%, respectivamente. Es importante considerar que los modelos no cooperativos solo lograron mejores valores que los modelos cooperativos en los conjuntos de datos ACHE, DHFR, THR y pLC50-96h-zebrafish por un 7 %, 1 %, 2 % y 1 %, respectivamente.

Tabla 13. Habilidades de generalización (Q_{test}^2) obtenidos por los mejores modelos no cooperativos y cooperativos y los mejores modelos reportados en la literatura para los conjuntos de compuestos ecotoxicológicos

	pNOAEC	Platyurus	48H	96H	120H	132H
&Mayor QuBiLS-MAS (Cooperativo)	0.89*	0.96	0.96	0.95	0.96	0.90
&&Menor QuBiLS-MAS (Cooperativo)	0.79	0.89	0.90	0.92	0.87	0.83
&&&promedio QuBiLS-MAS (Cooperativo)	0.86 (.023) **	0.94 (.060)	0.95 (.004)	0.94 (.008)	0.94 (.016)	0.88 (.015)
#mayor QuBiLS-MAS (No cooperativo)	0.86	0.94	0.95	0.96	0.93	0.78
##menor QuBiLS-MAS (No cooperativo)	0.70	0.89	0.87	0.92	0.87	0.70
###promedio QuBiLS-MAS (No cooperativo)	0.81 (.065)	0.92 (.014)	0.93 (.007)	0.95 (.010)	0.91 (.016)	0.73 (.023)
Modelo 1 Nath et al. (Sección 3.1 en (Nath et al., 2022))	0.681					
PLS Lavado et al. (Tabla 1 en (Lavado et al., 2021))		0.739				
Liu et al. (Tabla 2 en (T. Liu et al., 2020))			0.87	0.9237	0.8508	<u>0.8127</u>

*: El mejor valor por conjunto está marcado en negritas, mientras que el segundo mejor este subrayado

** : La desviación estándar del rendimiento de los mejores 30 modelos se muestra entre paréntesis

4.2.3 Estudio estadístico basado en estimación bayesiana

El rendimiento en los conjuntos de prueba (Q_{test}^2) de los modelos cooperativos se comparó con el rendimiento de los mejores modelos reportados en la literatura respecto a los catorce conjuntos de datos, (ver Tabla 14). De igual forma los modelos construidos por las estrategias cooperativa y no cooperativa se compararon en la Tabla 15. Para esto se tuvieron en cuenta los valores de Q_{test}^2 más altos, promedio y más bajos obtenidos por los modelos construidos. Si la capacidad de generalización (Q_{test}^2) de los modelos cooperativos se compara con la capacidad de generalización de los mejores modelos reportados en la literatura con respecto a los catorce conjuntos de datos, se puede observar que incluso los desempeños más bajos obtenidos por los modelos cooperativos son mejores que los mejores modelos del estado del arte con una probabilidad superior al 97%. Se puede observar que en promedio los modelos construidos a partir de subconjuntos de DMs optimizados a través de la estrategia cooperativa tienen 100% de probabilidad de ser superiores a los modelos que se construyen a partir de conjunto de DMs sin optimizar. Una conclusión similar se obtiene al analizar el rendimiento de los mejores modelos cooperativos.

Finalmente, se comparan el rendimiento de los modelos cooperativos y no cooperativos (ver Tabla 15). Para ello, se compara el rendimiento de los mejores modelos cooperativos y no cooperativos entre sí. Del mismo modo, se lleva a cabo el estudio para los valores más bajos de rendimiento de cada uno de los modelos, así como para el rendimiento promedio. En la tabla se puede observar que los modelos

construidos a partir de los DMs retornados por el enfoque cooperativo presentan una mayor probabilidad de lograr mejores desempeños que los modelos creados a partir de los modelos no cooperativos.

Tabla 14. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos cooperativos respecto a los mejores modelos reportados en la literatura.

Cooperativo	literatura	$p(\text{cooperativo})^a$	$p(\text{Rope})^b$	$p(\text{literatura})^c$
Mejor	Mejor	1.00	0.00	0.00
Promedio	Mejor	1.00	0.00	0.00
Menor	Mejor	0.97	0.00	0.03

^a: probabilidad de que el modelo cooperative sea significativamente mejor

^b: probabilidad de que ambos seas similares

^c: probabilidad de que el mejor modelo de la literatura sea superior

Tabla 15. Resultado del análisis estadístico basado en estimación bayesiana, entre el rendimiento de los modelos no cooperativos respecto a los modelos cooperativos.

Cooperativo	No cooperativo	$p(\text{cooperativo})^a$	$p(\text{Rope})^b$	$p(\text{no cooperativo})^c$
Promedio	Promedio	0.928	0.053	0.019
Mejor	Mejor	0.846	0.089	0.065
Menor	Menor	0.896	0.101	0.003

^a: probabilidad de que el modelo cooperative sea significativamente mejor

^b: probabilidad de que ambos seas similares

^c: probabilidad de que modelo no cooperative sea superior

Por lo tanto, los resultados estadísticos indican que la estrategia cooperativa es una mejor alternativa que el enfoque aplicado actualmente (es decir, sobrecálculo de DM y luego selección) para determinar buenos conjuntos de DMs para el modelado QSAR. Esto se debe a que se pueden construir modelos con rendimientos notablemente superiores a partir de los conjuntos "óptimos" de DMs obtenidos a través de del algoritmo C-AExOp-DCS. Es importante considerar que la superioridad estadística del rendimiento de los modelos construidos se obtuvo utilizando solo MLR, que es más simple que varias de las técnicas estadísticas o de aprendizaje (p. ej., PLS, GBM, RF) utilizadas para construir los modelos reportados en la literatura. En consecuencia, el rendimiento superior obtenido se debió principalmente a la calidad de los conjuntos "óptimos" de DMs obtenidos a través de las estrategias propuestas. De igual forma se puede concluir que la estrategia cooperativa es una mejor alternativa que la estrategia no cooperativa para determinar subconjuntos óptimos de DMs, dado que modelos con mayor poder predictivo pueden ser construidos a partir de los subconjuntos de DMs retornados por la estrategia cooperativa.

4.2.4 Análisis de los mejores modelos por cada conjunto de datos

La Tabla 16 muestra los valores de rendimiento de los mejores modelos cooperativos en cada conjunto de compuestos químicos estudiados. El coeficiente de determinación (R^2), el coeficiente de determinación ajustado (R_{adj}^2), el coeficiente de determinación con validación cruzada con 10 pliegues (R_{10-cv}^2), el error absoluto medio con validación cruzada (MAE_{10-cv}), el error cuadrático medio con validación cruzada ($RMSE_{10-cv}$), y el coeficiente de determinación de validación cruzada con valores de actividad perturbados ($R_{y-scramb}^2$) fueron las métricas de rendimiento utilizadas en el conjunto de datos de entrenamiento. Las técnicas de validación cruzada en 10 pliegues (J. H. Kim, 2009) y *Y-scrambling* (Rücker et al., 2007) se repitieron cada una 100 veces para determinar los valores de R_{10-cv}^2 , MAE_{10-cv} , $RMSE_{10-cv}$ y $R_{y-scramb}^2$, respectivamente. Además, las métricas Q_{test}^2 , MAE_{test} y $RMSE_{test}$ fueron las métricas calculadas para evaluar la capacidad de generalización de los modelos. Es importante señalar que los modelos analizados en esta sección estarán disponibles en las herramientas computacionales SiLiS-PREENZA (Yovani Marrero Ponce, 2023a) y SiLiS-PTOXRA (Yovani Marrero Ponce, 2023b), respectivamente.

Como un resultado, se puede observar en la Tabla 16 como todos los modelos cooperativos obtuvieron buenas métricas de desempeño en el conjunto de entrenamiento, presentando en su mayoría valores de R^2 , R_{adj}^2 y R_{10-cv}^2 entre 0.7 y 0.9. Solo los modelos para predecir pIC50-BZR, pIC50-COX2 y pLC50-132h-zebrafish obtuvieron valores R_{10-cv}^2 ligeramente inferiores a 0,7. Nótese además que los valores de Q_{test}^2 indican que todos los modelos lograron buenas habilidades de generalización. Todos los modelos obtuvieron valores de Q_{test}^2 superiores a 0.73, con excepción de los modelos para predecir pIC50-BZR y pIC50-COX2. No obstante, es importante señalar que los modelos cooperativos para predecir pIC50-BZR y pIC50-COX2 presentaron los mejores valores de Q_{test}^2 reportados hasta la fecha. En general, el rendimiento de estos modelos sugiere un buen balance bias-varianza ya que no existe diferencia entre los valores de R_{10-cv}^2 y Q_{test}^2 . Además, se puede observar que los valores de $R_{y-scramb}^2$ siempre fueron inferiores a 0,06, lo que indica que los DMs utilizados en estos modelos no se correlacionan aleatoriamente con las actividades modeladas y, por lo tanto, el buen desempeño de ellos no es casuístico.

Por último, la Tabla 16 muestra además la habilidad de generalización de cada modelo cooperativo dentro de su dominio de aplicabilidad (DA). Para determinar el DA de cada modelo se aplicó un enfoque basado en consenso (García-Jacas et al., 2019; Mora et al., 2020). Por un lado, si se analiza la cobertura de cada modelo (es decir, la proporción de compuestos que se espera que caiga dentro de la DA) se puede observar que los modelos para predecir pIC50-COX2, pIC50-DHFR, pLC50-48h-zebrafish, pLC50-96h-

zebrafish, pLC50-120h-zebrafish, pNOAEC, and pLC50-T.platyurus alcanzaron valores de cobertura superiores a 0,92, lo que implica que más del 92% de las predicciones realizadas por ellos son fiables. El resto de los modelos obtuvieron valores de cobertura entre 0,80 y 0,89, las cuales pueden considerarse adecuadas. Por otro lado, si se analizan las métricas de desempeño externo, se puede notar que todos los modelos muestran resultados comparables con respecto a los logrados cuando no se tuvo en cuenta el DA. Esto sugiere que los modelos creados mantienen buena capacidad de generalización dentro de su DA.

4.3 Conclusiones parciales

En este capítulo se examinó la calidad de los subconjuntos de DMs “optimizados” mediante el algoritmo C-AExOp-DCS en la construcción de modelos para predecir diferentes actividades biológicas en catorce conjuntos de compuestos químicos. Para esto, una vez ejecutado el algoritmo para cada uno de los conjuntos de compuestos y determinado los respectivos subconjuntos de DMs optimizados, se procedió a la construcción de diferentes modelos QSAR utilizando Regresión lineal múltiple como técnica estadística. Los resultados fueron comparados con los modelos disponibles en la literatura tomando en cuenta rendimiento observado tanto en el conjunto de entrenamiento como en los conjuntos de prueba.

Por lo general, se observó que los modelos cooperativos tuvieron mejor desempeño tanto en bondad de ajuste como en habilidades de generalización que los modelos reportados. De igual forma, se pudo observar cómo entre ambos algoritmos, los modelos cooperativos tienden a tener mejor habilidad de generalización que los los modelos no cooperativos. Adicionalmente, se validaron estadísticamente los valores de Q_{ext}^2 de los modelos construidos respecto a los mejores modelos del estado del arte y entre los modelos cooperativos y no cooperativos. De esta manera se pudo observar cómo los modelos construidos a partir de los DMs devueltos por el algoritmo cooperativo tienen mayor probabilidad de tener rendimiento superior a los mejores modelos del estado del arte. Además, se pudo observar cómo los modelos construidos a partir de los DMs obtenidos a partir del enfoque cooperativo presenta mayor probabilidad de tener rendimiento superior que los modelos construidos a partir del enfoque no cooperativo.

Tabla 16. Valores de rendimiento en entrenamiento, validación y DA alcanzados por los mejores modelos cooperativos en cada conjunto de datos.

Conjunto de datos	Conjunto de entrenamiento						Conjunto de prueba ^c		
	R^2	R^2_{adj}	R^2_{10-cv} ^a	MAE_{10-cv} ^a	$RMSE_{10-cv}$ ^a	$R^2_{y-scramb}$ ^b	Q^2_{test}	MAE_{test}	$RMSE_{test}$
pLC ₅₀ -ACE	0.8313	0.8193	0.8197 (.002)	0.8106 (.006)	0.9942 (.005)	0.0333 (.048)	0.7908 0.768(5)	1.2417 1.2181	1.4692 1.4392
pLC ₅₀ -ACHE	0.8318	0.8140	0.8150 (.003)	0.4192 (.005)	0.5208 (.004)	0.0217 (.027)	0.7335 0.7764(4)	0.5375 0.4671	0.6898 0.5795
pLC ₅₀ -BZR	0.7340	0.7100	0.7119 (.004)	.2841 (.003)	0.3551 (.002)	0.0224 (.029)	0.6892 0.6669(5)	0.5160 0.5065	0.6215 0.5900
pLC ₅₀ -COX2	0.6793	0.6650	0.6640 (.004)	0.4715 (.003)	0.5902 (.003)	0.0133 (.017)	0.5828 0.5730(5)	0.7667 0.7178	1.0415 0.9840
pLC ₅₀ -DHFR	0.7484	0.7395	0.7370 (.003)	0.5119 (.004)	0.6495 (.004)	0.0096 (.015)	0.7545 0.7530(5)	0.5536 0.5400	0.6754 0.6626
pK _i -GPB	0.8393	0.8081	0.8218 (.004)	0.3549 (.007)	0.4527 (.005)	0.0428 (.047)	0.8732 0.7263(4)	0.4061 0.3645	0.4989 0.4443
pK _i -THERM	0.8300	0.8068	0.8166 (.008)	0.6206 (.016)	0.8056 (.018)	0.0552 (.082)	0.7893 0.7801(5)	1.0414 0.9419	1.1724 1.0723
pK _i -THR	0.8557	0.8326	0.8344 (.002)	0.2969 (.004)	0.3877 (.003)	0.0439 (.063)	0.8301 0.7886(3)	0.4110 0.4066	0.5466 0.5540
pLC ₅₀ -48h-zebrafish	0.8508	0.8286	0.8334 (.003)	0.4606 (.006)	0.6137 (.005)	0.0072 (.011)	0.9592 0.9544(1)	0.2833 0.2854	0.3635 0.3668
pLC ₅₀ -96h-zebrafish	0.8686	0.8341	0.8446 (.003)	0.5123 (.009)	0.6416 (.006)	0.0318 (.043)	0.9538 0.9384(1)	0.4741 0.4952	0.5521 0.5703
pLC ₅₀ -120h-zebrafish	0.8409	0.8085	0.8228 (.002)	0.4180 (.004)	0.5429 (.003)	0.0099 (.013)	0.9507 0.9480(1)	0.3179 0.3053	0.4667 0.4575
pLC ₅₀ -132h-zebrafish	0.8086	0.7674	0.7804 (.003)	0.3038 (.004)	0.3899 (.003)	0.0161 (.021)	0.8715 0.7275(5)	0.3019 0.2849	0.3655 0.3534
pNOAEC	0.8234	0.8074	0.8112 (.002)	0.4542 (.005)	0.5889 (.003)	0.0245 (.031)	0.8701 0.8847(1)	0.3445 0.3283	0.4264 0.4069
pLC ₅₀ -T.platyurus	0.8371	0.8094	0.8102 (.004)	0.3460 (.005)	0.4359 (.005)	0.0265 (.034)	0.9546 0.9549(1)	0.2081 0.2155	0.2468 0.2536

^a: promedio luego de ejecutar 100 veces validación cruzada con 10 pliegues. La desviación estándar se muestra entre paréntesis.

^b: R^2_{10-cv} promedio de 100 ejecuciones del modelo original después de perturbar los valores de la actividad. La desviación estándar se muestra entre paréntesis.

^c: La segunda fila por modelo es el rendimiento del modelo considerando los compuestos dentro del dominio de aplicabilidad. Entre paréntesis se muestra el número de compuestos que no caen dentro.

Capítulo 5. Desarrollo de modelos para la predicción de actividad hepatotóxica

Un evento de lesión hepática inducida por medicamentos (DILI, por sus siglas en inglés) ocurre cuando una gran cantidad de hepatocitos (células hepáticas) se dañan químicamente (Nguyen-Vo et al., 2020). DILI es una de las principales causas del fracaso en el desarrollo de fármacos, también es una de las razones fundamentales para retirar fármacos del mercado (Mora et al., 2020). En consecuencia, un paso crítico en el proceso de desarrollo de fármacos es la identificación temprana de compuestos que podrían causar DILI. Identificar compuestos con propiedades hepatotóxicas puede ser extremadamente difícil considerando la diversidad de razones que pueden causarlo (Schyman et al., 2017). Por ejemplo, los compuestos con actividad DILI conocida en humanos pueden no mostrar una lesión hepática clara cuando se prueban en estudios con animales (Schyman et al., 2017). Además, existe el riesgo de encontrar compuestos que causen DILI solo en casos raros (idiosincrático) (Vall et al., 2021). Esta lesión no depende de la dosis y no es reproducible en estudios con animales (Vall et al., 2021).

Dada la relevancia que tienen las lesiones hepáticas, su impacto negativo en el proceso de desarrollo de fármacos y atendiendo a los resultados prometedores obtenidos por los modelos cooperativos, este capítulo tiene como objetivo la construcción de modelos para la predicción de DILI, a partir de subconjuntos “óptimos” de DMs retornados por la estrategia cooperativa presentada en el Capítulo 3. Para esto se usaron diez conjuntos de compuestos químicos utilizados en la literatura para la construcción y validación de modelos para la predicción de DILI.

5.1 Bases para la predicción de actividad hepática

En este estudio se utilizó un total de diez conjuntos, los cuales están descritos en la Tabla 17. Como conjuntos de entrenamiento se utilizaron tres conjuntos distintos: Liew_0_TR_1075 (Liew et al., 2011), Nguyen_0_TR_1596 (Nguyen-Vo et al., 2020) y Schyman_0_TR_1423 (Schyman et al., 2017). Para validar la habilidad de generalización de los modelos creados siete conjuntos de pruebas fueron tenidos en cuenta: Liew_1_TS_120 (Liew et al., 2011), Liew_2_B_TS_47 (Liew et al., 2011), Liew_3_ValPair_20 (Liew et al., 2011), Mora_4_ETS_554 (Mora et al., 2020), Garcia_1_TS_106, Nguyen_1_TS_322 (Nguyen-Vo et al., 2020) y Nguyen_2_TS_52. Los conjuntos de prueba Liew_1_TS_120 (Liew et al., 2011), Liew_2_B_TS_47

(Liew et al., 2011) y Liew_3_ValPair_20 (Liew et al., 2011) se han utilizado ampliamente en la literatura para medir la calidad de varios modelos. En particular, es importante resaltar el conjunto de datos Liew_3_ValPair_20, que está formado por 10 pares de compuestos estructuralmente similares, pero con diferente actividad de toxicidad (Liew et al., 2011).

El conjunto de prueba Mora_4_ETS_554 (Mora et al., 2020) es la unión de varios conjuntos de prueba ya reportados en la literatura sin compuestos duplicados respecto a Liew_0_TR_1075. Garcia_1_TS_106 se define en este trabajo a partir de los compuestos existentes en Mora_4_ETS_554, pero eliminando los compuestos duplicados con el conjunto de entrenamiento Schyman_1_TS_1423. Finalmente, también se consideró el conjunto de prueba Nguyen_1_TS_322 reportado en (Nguyen-Vo et al., 2020). Como este conjunto de prueba tiene compuestos duplicados con el conjunto de entrenamiento Nguyen_0_TR_1596, se construyó el conjunto de prueba Nguyen_2_TS_52, con los compuestos existentes en Nguyen_1_TS_322 pero eliminando duplicados con Nguyen_0_TR_1596. Este último conjunto se crea para medir verdaderamente las capacidades de generalización de los modelos creados a partir del conjunto de entrenamiento Nguyen_0_TR_1596.

Tabla 17. Conjuntos de datos utilizados para la construcción de modelos para la predicción de DILI

	código	cantidad positivo-DILI	cantidad negativo-DILI
<i>Conjuntos de entrenamiento</i>	Liew_0_TR_1075[5]	652	423
	Nguyen_0_TR_1596[51]	946	651
	Schyman_0_TR_1423[52]	779	644
<i>Conjuntos de prueba</i>	Garcia_1_TS_106	68	38
	Liew_1_R_TS_120[5]	72	48
	Liew_2_B_TS_47[5]	23	24
	Liew_3_ValPair_20[5]	10	10
	Mora_4_ETS_554[9]	376	178
	Nguyen_1_TS_322[51]	128	194
	Nguyen_2_TS_52[51]	25	27

^a: EL nombre al inicio del código hace referencia al primer autor del trabajo donde se reporta por primera vez el conjunto de datos y el número al final corresponde con la cantidad de compuestos del conjunto.

5.2 Procedimientos para construir y validar los modelos para la predicción de DILI

Cada uno de los tres conjuntos de entrenamiento se utilizó como entrada del algoritmo cooperativo C-AExOp-DCS. Como resultado, se obtuvieron tres subconjuntos con 172, 168 y 279 DMs relevantes para los conjuntos de entrenamiento de Liew_0_TR_1075, Schyman_0_TR_1423y Nguyen_0_TR_1596, respectivamente. A partir de cada subconjunto devuelto, se construyeron varios modelos QSAR utilizando

su respectivo conjunto de entrenamiento. Por cada conjunto de entrenamiento se seleccionaron los tres mejores modelos cooperativos según el valor de la correlación de Matthew, y su rendimiento se comparó con los modelos reportados en la literatura para cada conjunto de entrenamiento.

Por otro lado, para evaluar la habilidad de generalización de los modelos construidos se utilizaron siete conjuntos de prueba (consulte la Tabla 17). Los resultados se compararon con varios modelos y herramientas computacionales reportados en la literatura para predecir DILI (Bolón-Canedo & Alonso-Betanzos, 2019; Dong et al., 2018; He et al., 2013; Kang & Kang, 2021; T. Li et al., 2021; X. Li et al., 2018; Liew et al., 2011; Nguyen-Vo et al., 2020; Pires et al., 2015; Schyman et al., 2017; Xiong et al., 2021; Xu et al., 2015). Específicamente se analizaron los cuatro modelos reportados en (Mora et al., 2020). Dos de ellos están basados en *Random Forest* y se denotan como M2 y M9, mientras los otros dos modelos son basados en técnica de fusión Stacking y se denotan como E12 y E13. Hasta la fecha, estos modelos presentan el mejor desempeño para los conjuntos de Liew (Liew et al., 2011). Además, se consideraron los modelos presentados por Liew (Liew et al., 2011). También se consideró el rendimiento reportado por el modelo vslead(X. Li et al., 2018), el cual se construyó con una máquina de soporte vectorial utilizando *MACCS fingerprints*. También se tuvo en cuenta el modelo DNN-ECFP4 (Kang & Kang, 2021), construido utilizando una red neuronal profunda con *fingerprints* de conectividad extendida de diámetro 4 (EFCP4).

Además, se analizaron la predicción de varias herramientas y servidores como Dili-Server (Xu et al., 2015), Padel-DDPredictor (He et al., 2013), pkCSM (Pires et al., 2015), ADMETlab (Dong et al., 2018), ADMETlab 2.0 (Xiong et al., 2021), vNN-ADMET (Schyman et al., 2017), DeepDILI (T. Li et al., 2021) y DILI-CNN-MFE (Nguyen-Vo et al., 2020). Dili-Server (Xu et al., 2015) presenta dos modelos basados en aprendizaje profundo (Dili-Liew, Dili-Combined) construidos sobre el conjunto de entrenamiento de Liew. Padel-DDPredictor (He et al., 2013) proporciona un modelo basado en los modelos reportados por Liew (Liew et al., 2011). El servidor pkCSM proporciona modelos para predecir diferentes ADMET *endpoints*, entre ellos DILI, construidos a partir de técnicas basadas en grafos (Pires et al., 2015). De manera similar, ADMETlab proporciona varios modelos por ensemble para predecir DILI (Dong et al., 2018), mientras que su segunda versión, ADMETlab 2.0, proporciona una red neuronal basada en grafos (Xiong et al., 2021). La plataforma vNN-ADMET proporciona un modelo para la predicción DILI basado en la metodología del vecino más cercano variable (VNN) [52], construido a partir del conjunto de entrenamiento Scyman_0_TR_1423.

Finalmente, DeepDILI proporciona un modelo de aprendizaje profundo utilizando los DMs Mold2 (T. Li et al., 2021). DILI-CNN-MFE proporciona un modelo basado en redes neuronales convolucionales

utilizando *fingerprints* embebidos como DMs (Nguyen-Vo et al., 2020). En el caso de las herramientas DeepDILI y DILI-CNN-MFE descargamos su código fuente desde GitHub y las ejecutamos usando las configuraciones por omisión. Para DILI-CNN-MFE se consideraron las predicciones de los modelos construidos en las épocas 10, 11, 12 y 13 (DILI-CNN-MFE-(I-IV)) como se reporta en (Nguyen-Vo et al., 2020). Es importante señalar que los modelos DILI-CNN-MFE se construyen utilizando el conjunto de entrenamiento Nguyen_0_TR_1596.

5.3 Rendimiento de los modelos construidos en el conjunto de entrenamiento

La Tabla 18, muestra el rendimiento en el conjunto de entrenamiento de los mejores tres modelos cooperativos (L1, L2, L3) construidos a partir del conjunto de entrenamiento de Liew, así como los mejores modelos reportados en la literatura construidos a partir del mismo conjunto de entrenamiento (Liew et al., 2011; Mora et al., 2020; Xu et al., 2015). Para los tres modelos construidos se muestra el promedio de los valores de las métricas de evaluación luego de ejecutar 100 veces una validación cruzada en 10 pliegues. Para este conjunto de datos de entrenamiento, el modelo con mayor rendimiento reportado es el modelo por ensemble reportado por Liew (Liew et al., 2011). Es importante tener en cuenta que los valores reportados para este modelo no incluyen ninguna técnica de validación, por lo que el rendimiento reportado puede ser engañoso. Otros dos modelos por ensemble, E12 (Mora et al., 2020) y E13 (Mora et al., 2020), son los modelos con mayor rendimiento considerando una estrategia de validación (Mora et al., 2020). Estos modelos son un *ensemble* de 5 y 7 modelos bases, respectivamente.

Teniendo en cuenta los modelos base, los modelos L1, L2 y L3 lograron el mayor rendimiento, siendo la primera vez que un modelo base logra un valor de coeficiente de correlación de Matthews (MCC) de 0,5 o superior. Los modelos base con mejor rendimiento en el conjunto de entrenamiento de Liew reportados en la literatura son los modelos M2 y M9 (Mora et al., 2020). Respecto a estos modelos, los modelos L1, L2 y L3 son un 3 % mejor en cuanto a la métrica de precisión (ACC), mientras que si se considera la métrica MCC, los modelos construidos son un 5 %, 4 % y un 3 % mejores que el modelo M9, y un 7 %, 6 % y 5% que el modelo M2, respectivamente. Si se compara con el modelo DL-Liew, el cual se basa en aprendizaje profundo (Xu et al., 2015), los modelos cooperativos son un 12% y 11 % mejores respecto a la métrica MCC.

Tabla 18. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Liew_0_TR_1075

Modelo	Técnica de modelación	# DMs o Modelos bases	Liew_0_TR_1075 ^d			
			ACC	SEN	SPE	MCC
L1 ^a	Random Forest	25	0.77(.002)	0.88(.006)	0.61(.009)	0.52(.005)
L2 ^a	Random Forest	26	0.77(.002)	0.88(.006)	0.61(.008)	0.51(.005)
L3 ^a	Random Forest	26	0.77(.002)	0.88(.005)	0.61(.008)	0.50(.005)
M2 (Base)(Mora et al., 2020)	Random Forest	18	0.74	0.87	0.54	0.45
M9 (Base)(Mora et al., 2020)	Random Forest	18	0.74	0.76	0.71	0.47
E13 (Ensemble)(Mora et al., 2020)	Ensemble	5	<u>0.84</u>	<u>0.89</u>	<u>0.76</u>	<u>0.66</u>
E12 (Ensemble)(Mora et al., 2020)	Ensemble	7	0.83	0.88	<u>0.76</u>	0.65
Liew Ensemble ^b (Liew et al., 2011)	Ensemble	617	<u>0.88</u>	<u>0.92</u>	<u>0.81</u>	<u>0.74</u>
Liew Base ^c (Liew et al., 2011)	KNN(K=9)		0.68	0.67	0.70	0.36
DL-Liew (Dili server) (Xu et al., 2015)			0.70	0.70	0.70	0.39

^a: rendimiento promedio luego de repetir 100 veces la validación cruzada con 10 pliegues (R_{10-cv}^2). La desviación estándar se muestra entre paréntesis.

^b: Coeficiente de determinación sin aplicar validación cruzada (R^2)

^c: validación cruzada con 5 pliegues (R_{5-cv}^2)

^d: El valor más alto de cada métrica se destaca en negrita y subrayado, mientras que el segundo valor más alto solo se subraya.

Por otro lado, la Tabla 19 muestra el rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos (N1, N2, N3) construidos a partir del conjunto de entrenamiento de Nguyen (Nguyen-Vo et al., 2020). Para los tres modelos, se muestra el promedio de los valores de las métricas de evaluación luego de ejecutar 100 veces una validación cruzada en 10 pliegues. Para este conjunto no se ha reportado métricas de evaluación de entrenamiento. Como se puede ver los modelos construidos alcanzaron valores de precisión superiores al 79% así como valores de MCC superiores a 0,56. Estos valores son evidencia de modelos con buen desempeño.

Tabla 19. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Nguyen_0_TR_1596.

Modelo	Técnica de modelación	# DMs	Nguyen_0_TR_1596 ^b			
			ACC	SEN	SPE	MCC
N1 ^a	Random Forest	15	<u>0.80(.001)</u>	<u>0.87(.004)</u>	<u>0.68(.006)</u>	<u>0.57(.003)</u>
N2 ^a	Random Forest	16	<u>0.79(.002)</u>	<u>0.87(.004)</u>	<u>0.69(.007)</u>	<u>0.57(.004)</u>
N3 ^a	Random Forest	16	<u>0.79(.001)</u>	<u>0.87(.004)</u>	<u>0.68(.006)</u>	<u>0.56(.004)</u>

^a: rendimiento promedio luego de repetir 100 veces la validación cruzada con 10 pliegues (R_{10-cv}^2). La desviación estándar se muestra entre paréntesis.

^b: El valor más alto de cada métrica se destaca en negrita y subrayado, mientras que el segundo valor más alto solo se subraya.

Por último, la Tabla 20 muestra el rendimiento de entrenamiento de los tres mejores modelos cooperativos (S1, S2, S3) creados a partir del conjunto de entrenamiento de Schyman (Schyman et al., 2017). Para los tres modelos construidos, se muestra el promedio de los valores de las métricas de evaluación luego de ejecutar 100 veces una validación cruzada de 10 pliegues. En este conjunto de entrenamiento, los tres modelos construidos son superiores en un 2 % y un 3 % que el modelo propuesto por Schyman si la precisión es tomada en cuenta. Para este conjunto no se reporta valor de la métrica de MCC.

Tabla 20. Comparación del rendimiento en el conjunto de entrenamiento de los tres mejores modelos cooperativos respecto a los mejores modelos reportados en la literatura construidos sobre Schyman_0_TR_1423.

Modelo	Técnica de modelación	#DMs	Schyman_0_TR_1423 ^b			
			ACC	SEN	SPE	MCC
S1 ^a	Random Forest	<u>20</u>	<u>0.74(.002)</u>	<u>0.81(.005)</u>	<u>0.66(.006)</u>	<u>0.47(.004)</u>
S2 ^a	Random Forest	<u>22</u>	<u>0.74(.001)</u>	<u>0.81(.006)</u>	<u>0.66(.007)</u>	<u>0.47(.004)</u>
S3 ^a	Random Forest	<u>22</u>	<u>0.73(.002)</u>	<u>0.80(.005)</u>	<u>0.65(.006)</u>	<u>0.46(.006)</u>
<i>vNN web server</i> ^b (Schyman et al., 2017)	vNN		0.71	0.7	<u>0.73</u>	

^a: rendimiento promedio luego de repetir 100 veces la validación cruzada con 10 pliegues (R_{10-cv}^2). La desviación estándar se muestra entre paréntesis.

^b: El valor más alto de cada métrica se destaca en negrita y subrayado, mientras que el segundo valor más alto solo se subraya.

5.4 Capacidad de generalización de los modelos construidos

Como se mencionó anteriormente, la capacidad de generalización de los modelos construidos se determinó sobre siete conjuntos de datos. En este sentido, la **¡Error! No se encuentra el origen de la referencia.** muestra el valor de MCC obtenido por los nuevos modelos en cada uno de los conjuntos de prueba. Además, en la misma tabla se muestran los valores de MCC obtenidos por varios de los modelos y herramientas disponibles en la literatura para la predicción de DILI. Los valores para el resto de las métricas se pueden observar en las Tablas 22-25, así como los resultados del análisis del DA.

Si se analiza el rendimiento de los modelos para el conjunto de prueba Liew_1_R_TS_120, se puede observar cómo los modelos creados a partir del conjunto de entrenamiento Nguyen_0_TR_1596 y Schyman_0_TR_1423 lograron los mejores resultados. Los seis modelos alcanzan valores de MCC iguales o superiores a 0,90. Se puede observar cómo estos modelos lograron un mayor rendimiento que los modelos que se construyeron a partir del mismo conjunto de entrenamiento.

Tabla 21. Comparación del rendimiento según MCC en siete conjuntos de prueba de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI.

Modelo	Liew_1_R_TS_120	Liew_2_B_TS_47	Mora_4_ETS_554	Nguyen_1_TS_322	Nguyen_2_TS_52	Garcia_1_TS_106	Liew_3_ValPair_20
L1	0.56	0.71	0.50	0.74	0.73	0.53	0.12
L2	0.54	0.71	0.50	0.74	0.76	0.53	0.14
L3	0.54	0.71	0.50	0.75	0.73	0.58	0.11
N1	<u>0.93</u>	<u>1.00</u>	<u>0.75</u>	0.85	<u>0.86</u>	0.88	0.30
N2	0.90	<u>1.00</u>	<u>0.76</u>	0.85	<u>0.86</u>	<u>0.92</u>	<u>0.40</u>
N3	0.91	<u>1.00</u>	<u>0.75</u>	0.85	<u>0.86</u>	<u>0.92</u>	0.30
S1	<u>0.92</u>	<u>1.00</u>	0.64	0.87	<u>0.89</u>	0.82	0.11
S2	<u>0.92</u>	<u>1.00</u>	0.66	<u>0.88</u>	<u>0.89</u>	0.84	0.11
S3	<u>0.92</u>	<u>1.00</u>	0.64	<u>0.88</u>	<u>0.89</u>	<u>0.94</u>	0.11
M2 (Base)	0.53	0.76	0.46	0.64	0.70	0.58	0.12
M9 (Base)	0.51	0.71	0.45	0.54	0.56	0.69	0.00
E13 (Ensemble)	0.56	0.87	0.51	0.70	0.70	0.60	0.00
E12 (Ensemble)	0.56	0.87	0.53	0.71	0.68	0.60	0.00
Liew Ensemble	0.47	0.65	----	----	----	----	0.12
Liew Base	0.42	0.66	----	----	----	----	0.00
Vslead	0.63	0.47	0.50	----	----	----	----
Padel predictor	0.49	0.65	0.41	0.69	0.62	0.49	0.12
pkCSM	0.05	0.06	0.15	0.15	0.21	0.19	-0.12
AdmetLab	0.32	0.57	0.42	0.62	0.51	0.53	0.10
AdmetLab 2.0	0.26	0.58	0.38	0.65	0.66	0.54	0.00
vNN-Admet	0.76	0.91	0.60	<u>0.89</u>	<u>0.89</u>	0.71	0.10
Mold2+DeepDILI	0.27	0.87	0.43	0.74	0.51	0.62	0.00
DILI-CNN-MFE-I	0.52	0.57	0.57	0.62	0.61	0.78	0.10
DILI-CNN-MFE-II	0.72	0.84	0.69	0.75	0.67	0.88	0.20
DILI-CNN-MFE-III	0.71	<u>0.96</u>	0.69	0.77	0.68	0.88	<u>0.35</u>
DILI-CNN-MFE-IV	0.23	0.48	0.44	0.43	0.35	0.76	0.10

^a: El valor más alto por cada conjunto se destaca en negrita y subrayado, mientras que el segundo valor más alto solo se subraya.

Si analizamos los modelos construidos a partir del conjunto de entrenamiento Liew_0_TR_1075, se puede ver que el modelo L1 logró un rendimiento similar o superior al de los mejores modelos reportados construidos con el mismo conjunto de entrenamiento (E12 y E13) (Mora et al., 2020). Es importante señalar que estos modelos son construidos a partir de un ensemble de 5 y 7 modelos bases. Por otro lado, los modelos L2 y L3 tienen un rendimiento ligeramente inferior a estos modelos por ensemble, pero mayor que los mejores modelos bases reportados en la literatura (M2 y M9)(Mora et al., 2020). Respecto al resto de herramientas y modelos reportados se puede ver como los nueve modelos tienen mejor rendimiento, incluso cuando se compara con el rendimiento de servidores desplegados recientemente y construidos con técnicas basadas en aprendizaje profundo, tal como DeepDili (T. Li et al., 2021)

De igual forma se puede observar cómo los modelos N1, N2, N3, S1, S2 y S3 tienen el mejor rendimiento para el conjunto de prueba Liew_2_B_TS_45. Estos modelos lograron un rendimiento superior al de los modelos reportados creados a partir del mismo conjunto de datos de entrenamiento. No obstante, es importante remarcar que el traslape existente entre este conjunto de prueba y los conjuntos de entrenamientos para estos modelos es alto.

Entre los modelos construidos a partir del conjunto de entrenamiento de Liew, se puede observar cómo ambos modelos por ensemble E12 y E13 (Mora et al., 2020) alcanzaron los mejores resultados, teniendo un 16 % de superioridad respecto a los modelos cooperativos L1, L2 y L3. Entre los modelos bases, L1, L2 y L3 tuvieron un rendimiento inferior a M2 en un 5%, similar a M9 y superior al mejor modelo base de Liew en un 5%. Respecto al resto de los modelos reportados, los nueve modelos superan el rendimiento de los modelos pkCSM (Pires et al., 2015), DILI-CNN-MFE-(I, IV) (Nguyen-Vo et al., 2020), AdmetLab2.0 (Xiong et al., 2021), Liew ensemble (Liew et al., 2011), Liew base (Liew et al., 2011), Vslead (X. Li et al., 2018), PaDel predictor (He et al., 2013) y admetLab (Dong et al., 2018).

Si se analiza el rendimiento de los modelos para el conjunto de prueba Mora_4_ETS_554, se puede observar cómo los modelos N2, N3 y N1 alcanzaron los mejores resultados. Estos modelos son un 6 % y 7 % superior a los modelos DILI-CNN-MFE-(III-II), un 18 % y un 19 % superiores al modelo DILI-CNN-MFE-I, y un 31% y 32% superiores al modelo DILI-CNN-MFE-IV. Estos modelos profundos DILI-CNN-MFE(I-IV) y los modelos N1, N2 y N3 se construyeron sobre el mismo conjunto de entrenamiento. Por otro lado, los modelos S2, S3 y S1 fueron superiores al modelo de vNN-Admet (Schyman et al., 2017) por un 6%, 4% y 4%, respectivamente.

Si se comparan los modelos construidos a partir del conjunto de entrenamiento de Liew, se puede

observar cómo los modelos por ensemble E12 y E13 (Mora et al., 2020) fueron superiores por un 3% y 1% que los modelos cooperativos. Los modelos L1, L2 y L3 tuvieron un rendimiento superior en un 4% y 5% que los modelos bases M2 y M9. De igual forma se puede observar cómo los nueve modelos tuvieron un rendimiento superior al resto de modelos y herramientas.

Por otro lado, si los modelos se comparan analizando el conjunto de pruebas Nguyen_1_TS_322 se puede observar que el mejor modelo es vNN-Admet (Schyman et al., 2017), el cual es ligeramente superior a los modelos cooperativos construidos sobre el mismo conjunto de entrenamiento por un 1% y 2%. Si se analizan los modelos construidos a partir del conjunto de entrenamiento de Nguyen, los tres modelos cooperativos fueron superiores por un 8% y 10% a los modelos DILI-CNN-MFE-(III-II) (Nguyen-Vo et al., 2020), mientras que fueron al menos un 20% superior a los modelos DILI-CNN-MFE-(I-IV) (Nguyen-Vo et al., 2020).

Si se analizan los modelos construidos a partir de Liew, se puede observar cómo L3 fue ligeramente superior al modelo de DeepDILI (T. Li et al., 2021), mientras que L1 y L2 tuvieron rendimientos similares a este último. Se puede ver cómo los modelos L1, L2 y L3, son superiores al resto de los modelos considerados en al menos un 4%, incluyendo a los modelos por ensamble E12 y E13 (Mora et al., 2020), así como a los modelos construidos a partir de aprendizaje profundo.

Como se ha mencionado anteriormente, el conjunto de prueba Nguyen_1_TS_322 tiene solapamiento con el conjunto de entrenamiento del mismo autor Nguyen_0_TR_1596, por lo que el análisis comparativo entre los modelos que se construyen a partir de este conjunto de entrenamiento puede conducir a conclusiones erróneas. Para esto, se construyó el conjunto de pruebas Nguyen_2_TS_52, el cual tiene los compuestos existentes en Nguyen_1_TS_322 que no estén repetidos en el conjunto de entrenamiento. Para este conjunto, se puede observar como todos los modelos construidos sobre el conjunto de entrenamiento de Schyman (Schyman et al., 2017), incluyendo los modelos cooperativos S1, S2 y S3 tuvieron un rendimiento similar entre ellos y superior al resto de los modelos construidos, seguidos por el resto de los modelos cooperativos. Se puede ver en este conjunto de pruebas cómo todos los modelos cooperativos son superiores a cualquier otro modelo analizado.

De igual forma, con el objetivo de comparar los modelos construidos sobre el conjunto de entrenamiento de Schyman, y considerando que este conjunto tenía alto solapamiento con los conjuntos de prueba que utilizándose estaban usando, se construyó el conjunto de prueba Garcia_1_TS_106, el cual contiene los compuestos existentes en el conjunto de prueba de Mora (Mora et al., 2020), pero sin

compuestos duplicados con el conjunto de entrenamiento de Schyman. Para este conjunto de prueba el mejor modelo es el modelo cooperativo S3, el cual fue un 23% superior al modelo de vNN-Admet (Schyman et al., 2017), mientras que los otros modelos cooperativos construidos a partir del conjunto de entrenamiento de Schyman, S2 y S1, fueron de igual forma superiores a vNN-Admet por un 11% y 13%, respectivamente. Si los modelos construidos a partir del conjunto de entrenamiento de Nguyen (Nguyen-Vo et al., 2020) son analizados, se puede ver que los modelos cooperativos N2 y N3 fueron 4% superiores a los modelos DILI-CNN-MFE-(III-II) (Nguyen-Vo et al., 2020), mientras que N1 tiene igual valor de MCC que estos últimos. Si se comparan los modelos cooperativos N1, N2 y N3 respecto a los modelos DILI-CNN-MFE-(I-IV), los primeros fueron superiores en más de un 10%.

Finalmente, al analizar los modelos construidos sobre Liew, se puede ver como el mejor modelo fue M9, seguido de los dos modelos por ensemble E12 y E13 (Mora et al., 2020). Estos modelos son 11%, 2% y 2% superiores al modelo cooperativo L3, mientras que fueron 16%, 7% y 7% superiores a los modelos L1 y L2.

5.4.1 Análisis del rendimiento de los modelos en el conjunto de prueba Liew_3_ValPair_20

El rendimiento de los nueve modelos cooperativos es analizado teniendo en cuenta el conjunto de pruebas Liew_3_ValPair_20 (Liew et al., 2011) (ver Figuras 13 y 14). Este conjunto de datos es de gran importancia dada la complejidad de sus compuestos, ya que está formado por 10 parejas de compuestos similares estructuralmente, pero con diferentes actividades. Por ejemplo, en la figura se muestra como la *niacina* y el *ácido benzoico* tienen estructuras similares, pero al mismo tiempo presentan actividades distintas en cuanto a DILI. Para este conjunto el mejor modelo fue el modelo cooperativo N2, el cual tiene rendimiento superior en un 5%, 20% y 30% que los modelos DILI-CNN-MFE-(III, II, I, IV).

Entre los modelos construidos sobre el conjunto de entrenamiento de Liew, L2 es superior en 2% a el modelo M2, 14% superior al modelo M9 y a los modelos por ensemble E12 y E13 y a M9. Mientras que, si se analizan los modelos construidos sobre Schyman, se puede ver como los modelos cooperativos son ligeramente superiores al rendimiento del modelo de vNN-Admet. Observe cómo M9, DeepDILI, E12, E13 y Liew base tienen rendimiento igual cero, indicativo de que se está en presencia de modelos aleatorios, mientras que pkCSM (Pires et al., 2015) y DL-Liew (Xu et al., 2015) tienen rendimiento por debajo de cero.

5.5 Análisis de la diversidad y complejidad de los conjuntos analizados

Como se mencionó al inicio del Capítulo, identificar compuestos con propiedades hepatotóxicas puede ser extremadamente difícil, teniendo en cuenta la diversidad de razones que pueden causarlo (Schyman et al., 2017). En esta sección se realiza un análisis de la complejidad y diversidad de los conjuntos de compuestos utilizados con el objetivo de mostrar como los algoritmos de búsqueda propuestos en esta investigación mantienen una alta calidad en los conjuntos de DMs devueltos cuando aumenta la diversidad de los conjuntos y la complejidad de los compuestos. Para esto se realizan dos análisis. El primero es de diversidad basado en los *scaffold* químicos de los compuestos analizados (Saldívar-González & Medina-Franco, 2020), mientras que el segundo es un análisis de complejidad basado en una función de puntuación espacial (nSPS) (Krzyzanowski et al., 2023).

Análisis de diversidad: El análisis de diversidad se realizó vía la entropía de Shannon de la distribución de la cantidad de *scaffold* únicos en los conjuntos (Saldívar-González & Medina-Franco, 2020). Un *scaffold* molecular hace referencia a la estructura central de la molécula y es utilizado como punto de partida en el diseño de nuevos fármacos (Schaub et al., 2022). Para obtener el *scaffold* de cada compuesto utilizamos la implementación del algoritmo propuesto por Bemis y Murcko (Bemis & Murcko, 1996) disponible en el API para Java de CDK (Willighagen et al., 2017). Una vez determinado el *scaffold* para cada molécula se determinan las frecuencias de cada *uno* y sobre las frecuencias se determina la entropía. A mayor entropía más diverso es el conjunto de compuestos.

La Figura 7 **Error! No se encuentra el origen de la referencia.** muestra la entropía de Shannon normalizada de las frecuencias para los *scaffold* existentes en cada conjunto de datos. La entropía es normalizada respecto a la entropía máxima que puede ser alcanzada por cada distribución, teniendo en cuenta que los conjuntos de datos no tienen la misma cantidad de compuestos. En la figura se puede observar cómo los conjuntos más diversos son los conjuntos de compuestos de Liew y Nguyen, ambos con un valor de entropía de 0.84, seguidos por el conjunto de compuestos de Schyman con 0.83. El conjunto de compuestos utilizados para medir toxicidad en un ambiente acuático a partir de embriones de peces cebras luego de 132 horas de exposición es el siguiente conjunto más diverso con una entropía de 0.82. El resto de los conjuntos de compuestos presentan una entropía inferior a 0.8, siendo los conjuntos menos diversos los conjuntos de compuestos GPB y THER.

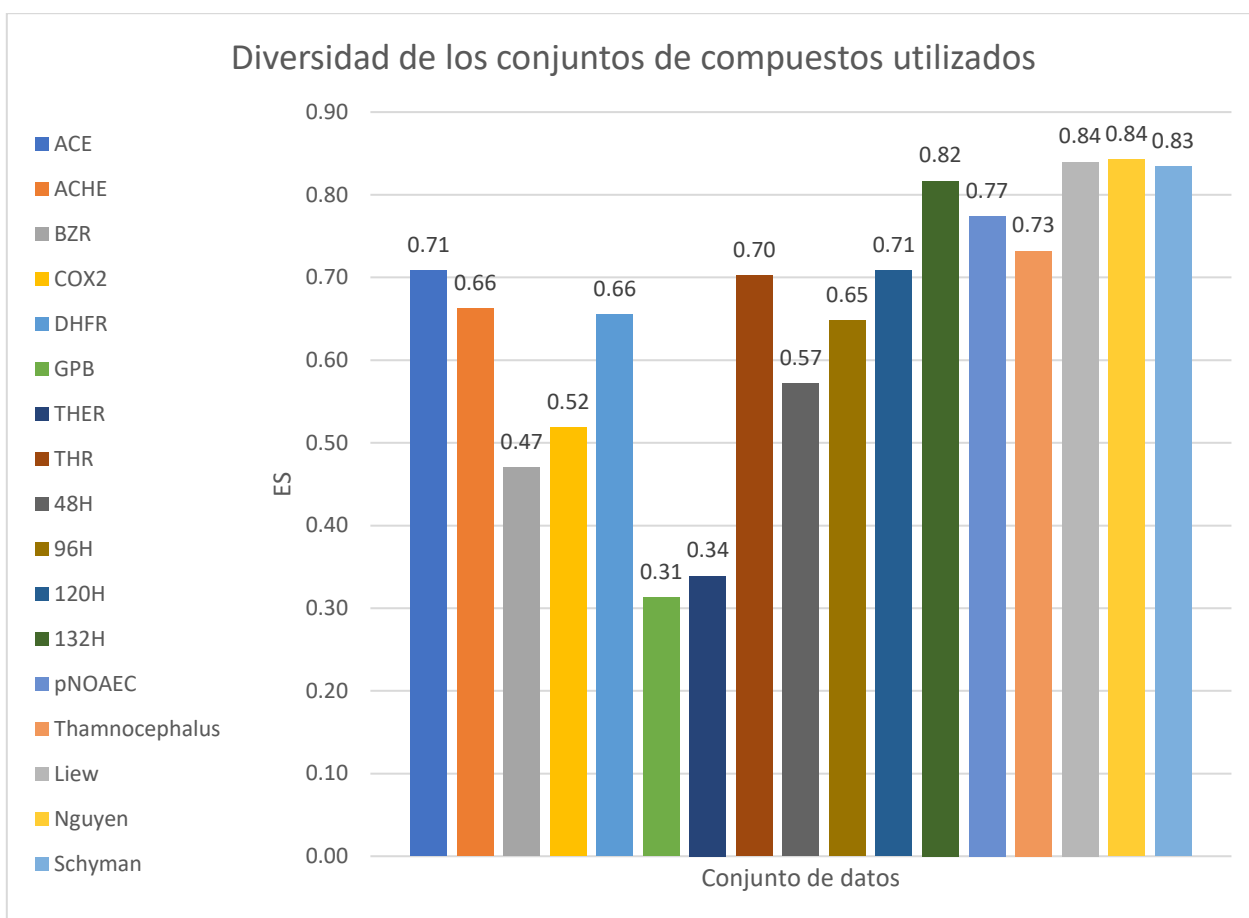


Figura 7. Entropía de Shannon normalizada de las frecuencias de los *scaffold* por cada conjunto de datos.

Análisis de complejidad: Para determinar la complejidad estructural de los compuestos que se están analizando se utilizó la función de puntaje espacial (SPS por sus siglas en inglés) (Krzyzanowski et al., 2023). Esta función es una integración de las funciones fracción de carbonos sp^3 -hibridados (F_{Sp}^3) y la fracción de carbonos estereogénicos ($F_{Cstereo}$) (Krzyzanowski et al., 2023). Para determinar el valor de SPS de una molécula se utiliza la siguiente función:

$$SPS = \sum_{a_i} h_{a_i} s_{a_i} r_{a_i} n_{a_i}^2 \quad (17)$$

donde a_i hace referencia a un átomo pesado. El término h_{a_i} hace referencia a los grados de libertad conformacional del átomo. Por otro lado, s_{a_i} es un término estereoisomérico, donde a los carbonos tetraédricos (pseudo) estereogénicos y a los átomos involucrados en un enlace doble con posibles

isómeros E y Z se les asigna $s = 2$, y en otros casos, el término es igual a 1. Por otro lado, r_{a_i} es igual a 2 cuando el átomo es parte de un anillo no aromático, en otro caso es 1. Por último, $n_{a_i}^2$ hace referencia a la cantidad de átomos pesados que comparten enlace con el átomo a_i .

La Figura 8 muestra la distribución de los valores de SPS de cada conjunto de compuestos estudiados. Se puede observar cómo los tres conjuntos de DILI tienen los compuestos más complejos, seguido por ACE, GPB y THR. Por otro lado, se puede observar además como los compuestos menos complejos están en el conjunto donde se encuentran los compuestos a los que se les han medido su toxicidad por inhalación (pNOAEC) y el conjunto donde se encuentran compuestos a los que se les han medido su toxicidad en un ambiente acuático, luego de 48 horas (48H) de exposición.

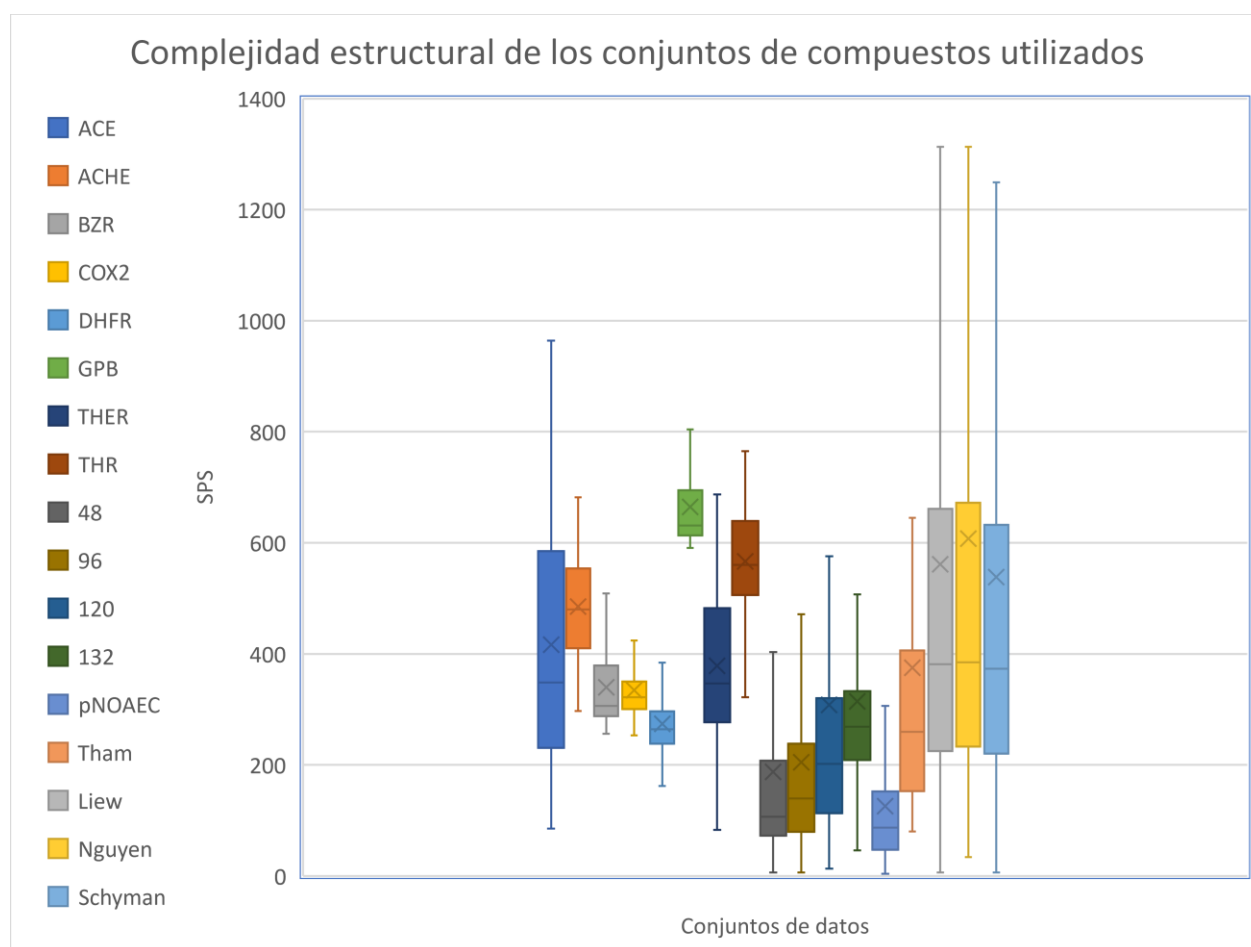


Figura 8. Puntaje espacial por cada conjunto de datos.

Se puede ver entonces que los conjuntos de DILI presentan mayor diversidad y complejidad que los restantes conjuntos de compuestos. Esto implica que la búsqueda de DMs capaces de caracterizar de

forma adecuada los compuestos en dichos conjuntos sea más difícil y, por lo tanto, más complejo construir modelos QSAR con un poder predictivo deseado. Por lo que se hace más importante definir que DMs son relevantes al problema que se quiera modelar. En este sentido, se pudo validar mediante la construcción de modelos QSAR, que los DMs retornados por la estrategia cooperativa son capaces de caracterizar en mejor medida el espacio químico que se está estudiando, incluso cuando aumenta la complejidad y la diversidad de los conjuntos químicos.

5.6 Conclusiones parciales

En este capítulo la calidad de nueve modelos para la predicción de DILI ha sido examinada. Estos modelos fueron construidos a partir de subconjuntos de DMs “optimizados” mediante la estrategia cooperativa propuesta en esta investigación. Para esto, una vez ejecutado el algoritmo cooperativo para cada uno de los conjuntos de compuestos de entrenamiento y determinado los respectivos subconjuntos de DMs optimizados, se procedió a la construcción de diferentes modelos QSAR. Los resultados obtenidos fueron comparados con los modelos disponibles en la literatura acorde al rendimiento observado tanto en conjunto de entrenamiento como en siete conjuntos de prueba.

En general, se observó que los rendimientos de los nueve modelos construidos tuvieron mejor desempeño tanto en bondad de ajuste como en habilidades de generalización que los modelos reportados. De igual forma, si se comparan cada uno de los modelos cooperativos respecto a los modelos reportados en la literatura construidos a partir del mismo conjunto de entrenamiento podemos observar cómo los modelos cooperativos tienen rendimientos superiores. Finalmente, se realizó un análisis de la complejidad y la diversidad de los compuestos bajo estudio, demostrándose que el enfoque cooperativo retorna DMs capaces de caracterizar de mejor forma los compuestos bajo estudio, incluso cuando los conjuntos son más diversos y complejos.

Capítulo 6. AExOp-DCS: Software para Explorar y Optimizar el Espacio de Configuración del Descriptor

En este capítulo se describen las principales características del software AExOp-DCS, herramienta que permite la ejecución de los algoritmos propuestos en la presente investigación. Se mencionan además las dependencias, ambiente de desarrollo, interfaces de usuario, arquitectura y pasos necesarios para su ejecución. Por otro lado, se describe la Interfaz de Programación Abstracta (API) y como usarla en otras herramientas y *pipelines*.

La Figura 9 muestra el diagrama básico del software AExOp-DCS. Este se puede ejecutar desde una interfaz de usuario basado en líneas de comandos (CLI) o una interfaz de usuario basada en interfaces gráficas (GUI). Para ambas interfaces es necesario definir cuatro parámetros de entrada: un archivo con el conjunto de moléculas en formato *sdf*, la actividad a predecir, un archivo con la configuración del algoritmo, y una dirección donde se guardará el mejor subconjunto. El archivo de configuración es definido mediante la interfaz gráfica. Es importante señalar que la actividad a predecir debe estar incluida en el fichero *sdf* como un atributo para cada una de las moléculas que se van a analizar. Cada una de las interfaces recibe los cuatro parámetros de entrada y envía a ejecutar el algoritmo AExOp-DCS a través del API desarrollada.

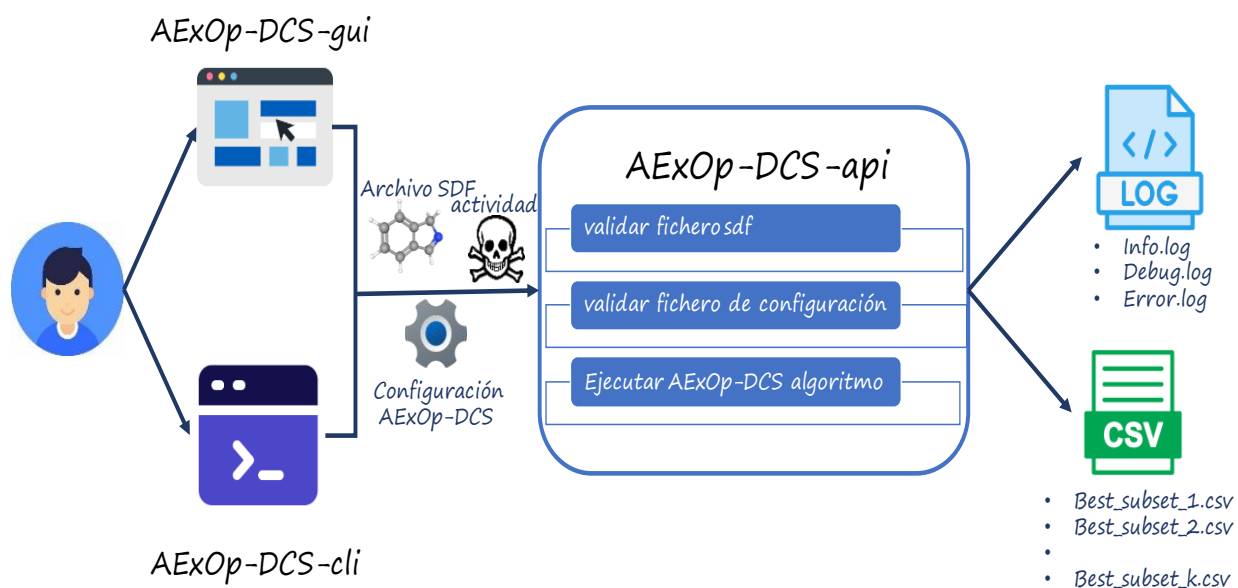


Figura 9. Diagrama básico del software AExOp-DCS

El API recibe los parámetros de entrada y valida que los archivos no estén corruptos y que la configuración definida sea correcta. Una vez validados los datos de entrada se comienza la ejecución del algoritmo. Para tener una bitácora de los eventos ocurridos durante la ejecución del algoritmo se escriben tres ficheros de registros o *logs*. El registro *info.log* contiene información básica sobre el estado de la ejecución. Por otro lado, el registro *debug.log* contiene información más detallada, por ejemplo, los descriptores seleccionados para realizar el operador de cruce en cada iteración o las mutaciones realizadas. Finalmente, en caso de que durante la ejecución se produzca un error, el registro *error.log* contendrá la información necesaria para determinar las causas por las que se produjo dicho error. Por otro lado, cada uno de los subconjuntos encontrados durante la ejecución del algoritmo será almacenado en un archivo con formato *csv*.

6.1 Dependencias y ambiente de trabajo

AExOp-DCS se desarrolló utilizando Java v1.8 (ORACLE. *Oracle Java*, 2023) como lenguaje de programación, IntelliJ IDEA 2022.2 como entorno de desarrollo integrado (IDE) (JETBRAINS. *IntelliJ IDEA*., 2023) y Apache Maven v3.8.6 como herramienta para la gestión y construcción de los proyectos (APACHE. *Apache Maven*, 2023). Además, se utilizó Chemistry Development Kit (CDK) v1.4.19 (Willighagen et al., 2017) para manipular las estructuras químicas, ToMoCoMDLibrary v2.0 para calcular los DMs QuBiLS-MAS (García-Jacas, Marrero-Ponce, Vivas-Reyes, et al., 2020) y el API de Weka v3.9.4 (WEKA software, s/f) para la manipulación de los datos y la implementación de la lógica de los algoritmos de regresión y clasificación.

6.2 Descripción de la Interfaz de usuario por la línea de comandos (CLI)

El módulo de interfaz de línea de comandos (CLI por sus siglas en inglés) permite ejecutar el algoritmo AExOp-DCS en una línea de comandos no interactiva. Este módulo puede ser útil si es necesario llamarlo desde scripts, *cronjobs*, terminales sin soporte para X-Windows, clústeres de computadoras, o dentro de un pipeline en otra herramienta. Además, puede ser útil si los usuarios prefieren ejecutar el algoritmo en segundo plano o cuando la longitud del conjunto de datos químicos implica un mayor tiempo de ejecución.

La **¡Error! No se encuentra el origen de la referencia.** muestra las opciones para ejecutar el software en modo consola. La CLI recibe cuatro parámetros obligatorios, la ruta donde se guardará el mejor

subconjunto, el nombre de la actividad o propiedad, la ruta del archivo del proyecto y la ruta del archivo *sdf*. Además, se habilita una opción opcional para imprimir los *logs* de la ejecución. Cuando se completa la ejecución, los *logs* se guardan en la carpeta de *logs* y el mejor subconjunto se guardará en la ruta definida en los argumentos.

```
$> java -jar AExOp-DCS.jar -h
usage: cmd [-c <arg>] [-d] [-e <arg>] [-h] [-i] [-p <arg>] [-s <arg>] [-v]
  -c,--csvfile <arg>    output, path to csv file
  -d,--debug            create logs folder for save application logs
  -e,--endpoint <arg>  property target
  -h,--help            Show this help and exit
  -i,--info            print project configuration
  -p,--project <arg>   path to project file .dcs
  -s,--sdffile <arg>  input, path to sdf file
  -v,--version         show the version and exit
```

Línea de comando 1. Menú de ayuda del software AExOp-DCS

6.3 Descripción de la Interfaz Gráfica de Usuario (*GUI*)

Se desarrolló una interfaz gráfica de usuario (*GUI* por sus siglas en inglés) para facilitar la configuración y la ejecución de la búsqueda. La interfaz proporciona una forma sencilla de configurar los parámetros de los algoritmos, tales como: operadores evolutivos, filtros, funciones de calidad y lista *DCSs*. Además, la interfaz permite generar nuevos proyectos de configuración, guardarlos, cargarlos y modificarlos.

La Figura 10 **¡Error! No se encuentra el origen de la referencia.** muestra la ventana principal de la interfaz de usuario. Esta interfaz tendrá dos funciones principales. La primera es construir el archivo de configuración necesario para la ejecución del algoritmo en ambas interfaces, mientras que la segunda función es lanzar la ejecución de los algoritmos. En la parte superior de la ventana se escribe la dirección del archivo *sdf*, la actividad a predecir y la ruta de salida del archivo *csv*. A la izquierda de la ventana se pueden definir los parámetros del algoritmo. En la pestaña "*Search strategy*" se pueden configurar los parámetros para el algoritmo de búsqueda, las funciones de calidad y los filtros. En la pestaña "*Search configuration*" se pueden definir los operadores genéticos, el número de iteraciones, el tamaño de la población y el operador de reinicio. Además, es posible decidir si se ejecutará el enfoque cooperativo o no. La pestaña "*Fitness function*" permite configurar la función para calcular la calidad de los *DMs* y la función para buscar y calcular la aptitud del subconjunto encontrado.

La pestaña "*Filter Function*" permite seleccionar y configurar las funciones de filtros para eliminar DMs no deseados. Por otro lado, la pestaña "*Molecular Descriptors*" permite configurar los DCSs que se quieren analizar. En esta pestaña se pueden seleccionar los valores deseados para cada uno de los parámetros de los que dependen los DCSs disponibles. Solo se han implementado los descriptores QuBiLS-MAS, por lo que los DCSs que se definan estarán relacionados a estos descriptores. Los DCSs se dividirán por cada forma algebraica (i.e., lineal, bilineal y cuadrática) usada en el cálculo de los DMs QuBiLS-MAS, teniendo en cuenta que estos no se pueden mezclar entre sí. De esta forma se pueden definir diferentes DCSs (espacios de configuración del descriptor) para una misma familia de DMs QuBiLS-MAS basada en una misma forma algebraica. Por ejemplo, se puede definir dos DCSs asociados a los DMs QuBiLS-MAS lineales, uno que esté basado en átomos y el otro basado en enlaces. Cada DCS definido se agregará usando el botón "*Add DCS*". En el centro de la aplicación mostramos la lista con los DCSs definidos.

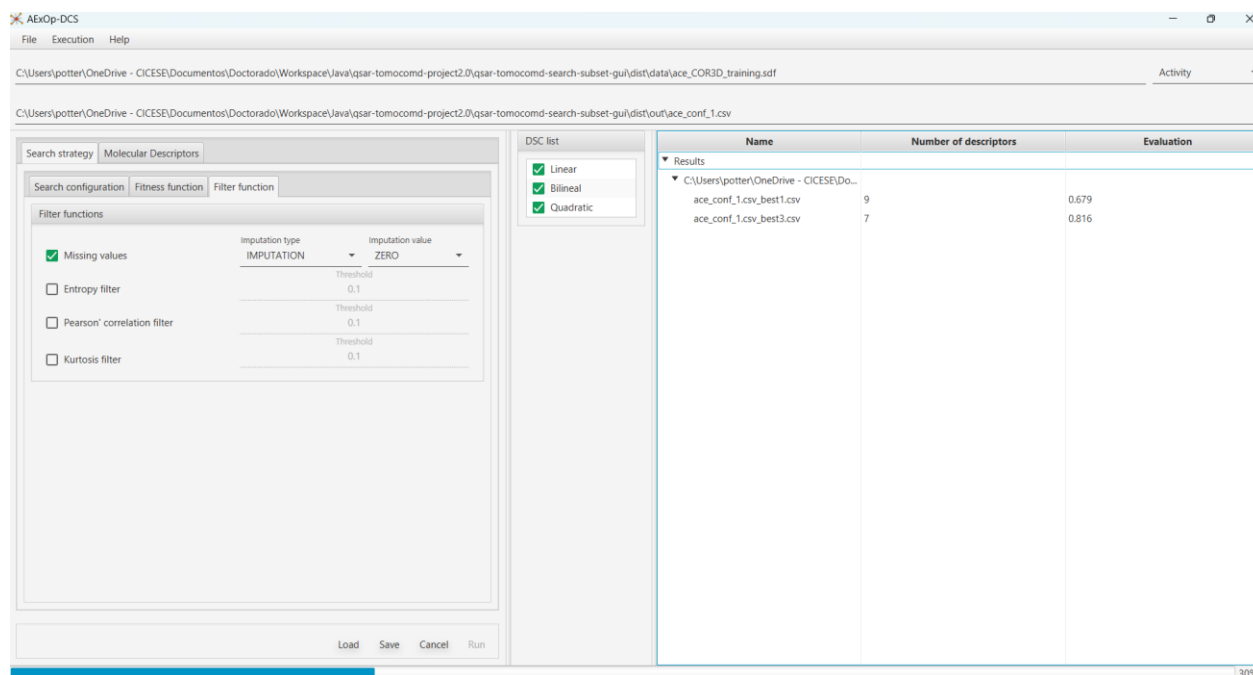


Figura 10. Ventana principal del software AExOp-DCS

Por otro lado, a la derecha de la ventana se encuentra la tabla de los resultados. Cada vez que el algoritmo encuentre un mejor subconjunto, en la tabla de los resultados se muestran la dirección donde se guardó el documento con los nuevos descriptores, la cantidad de descriptores en el subconjunto, y el valor de calidad del subconjunto. En la parte inferior de la ventana están los botones *Load*, *Save*, *Run* y *Cancel*, los cuales son para cargar un proyecto de configuración de la ejecución, guardar una nueva configuración, ejecutar la configuración, o cancelar una ejecución activa.

6.4 Interfaz de Programación Abstracta (API)

Como se mencionó, toda la lógica principal de los algoritmos propuestos se implementa dentro de una API. Este diseño garantiza la reutilización del código en otras herramientas y en futuras versiones del algoritmo. El API está organizado en los paquetes *data*, *exceptions*, *io*, *subsetsearch* y *utils*. En el paquete de *data* se implementa la clase que almacena la evaluación de los DMs en el conjunto de moléculas que se esté analizando. En el paquete *io* se definen las clases relacionadas con la lectura y escritura de los archivos necesarios para la ejecución, mientras que en el paquete *exceptions* se definen un conjunto de excepciones que pueden ocurrir durante la ejecución del algoritmo. Por último, en el paquete *utils* se definen clases para calcular diferentes operadores estadísticos necesarios en nuestra ejecución, además de contener la clase en la que se definen los parámetros de entrada de la interfaz de usuario por consola.

En el paquete *subsetsearch* están definidos los paquetes y clases necesarias para la ejecución principal del algoritmo (ver Figura 11). El mismo está compuesto por los paquetes *configuration*, *descriptors*, *evaluation*, *filters* and *search*. El paquete *configuration* (ver Figura 12) contiene las clases de configuración del algoritmo. En estas se definen los operadores que se utilizarán, así como sus parámetros y opciones. Por ejemplo, en la clase *GASelectionConfig* se almacena cuál será el operador de selección a utilizar durante el proceso evolutivo y sus opciones. Algo similar pasa con las clases *GAMutationConfig*, *GAMDRReplaceConfig*, *GACrossoverConfig* y *GAResetConfig* donde se definen los operadores de mutación, remplazo, cruzamiento y el reinicio de las poblaciones respectivamente.

La clase *MDEvaluationConf* es la encargada de almacenar la configuración de la función de calidad para los DMs, mientras que *FilterConfig* define los filtros con los cuales se eliminarán los DMs no deseados. La clase *GAAAlgorithm4PopConfig* define entonces todas las configuraciones necesarias para la ejecución de la parte evolutiva del algoritmo. Es en esta clase por ejemplo donde se define la dimensión de las poblaciones. Finalmente, la clase *GAConf* contiene toda la configuración necesaria para la ejecución del algoritmo completo.

La clase *LaunchExec* es la encargada de lanzar la ejecución (ver Figura 11). La clase *AGeneticAlgorithm* implementa la lógica de ambos algoritmos propuestos. Para esto se auxilia de las clases e interfaces *GAMDPopulation*, *IEvaluateSubset*, *IMDComputer* y *ResetPopulation*. La clase *GAMDPopulation* implementa la lógica del proceso evolutivo que se realiza por cada población. *IEvaluateSubset* es una interfaz donde se definen las funciones para determinar la calidad de los

subconjuntos, mientras que *IMDComputer* define la implementación de los algoritmos para el cálculo de descriptores moleculares. Por último., *ResetPopulation* define cómo se realizará el reinicio de las poblaciones.

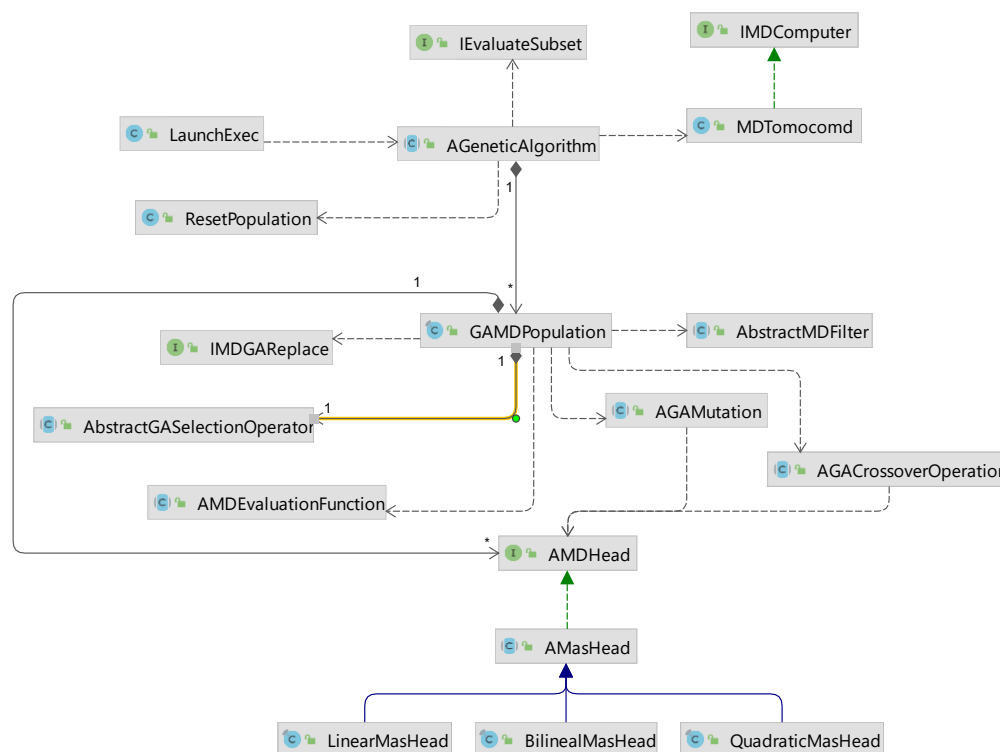


Figura 11. Diagrama UML para las clases en las que están implementada la lógica principal

Para ejecutar la parte evolutiva se tiene la clase *GAMDPopulation*. Para esto se definen además las interfaces y clases *IMDGARepalce*, *AbstractMDFilter*, *AbstractGASelectionOperator*, *AMDEvaluationFunction*, *AGAMutation*, *AGACrossoverOperation* y *AMDHead*. La clase *AMDHead* define la forma en que los cromosomas serán implementados. *AGAMutation* y *AGACrossoverOperation* implementan la lógica para los operadores de mutación y cruzamiento, respectivamente. *IMDGARepalce* es una interfaz que define la implementación para los operadores de remplazo, mientras que *AbstractMDFilter* implementa los filtros que se utilizan para eliminar los DMs no deseados. La clase *AbstractGASelectionOperator* define la lógica para los operadores de selección, mientras que *AMDEvaluationFunction* define la lógica para las funciones que determinan la calidad de los descriptores.

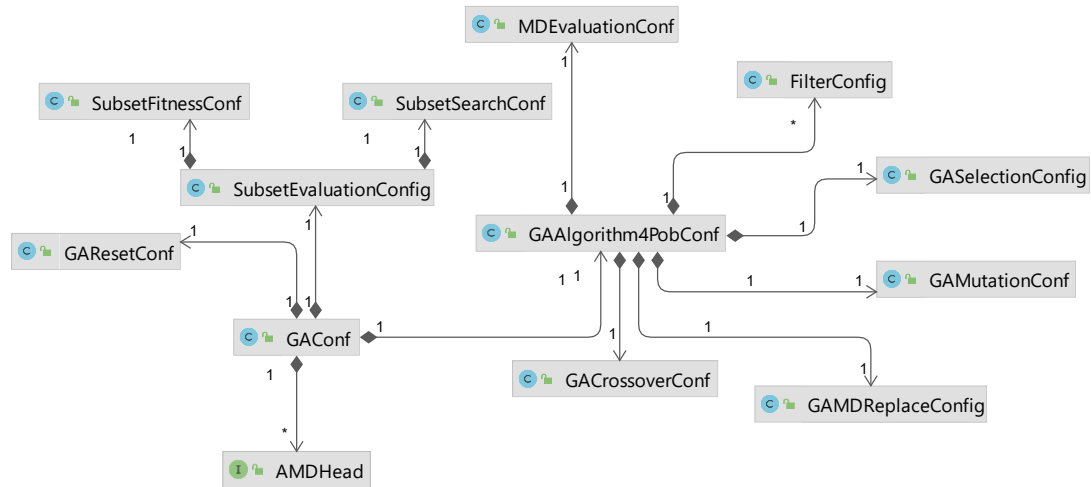


Figura 12. Diagrama UML para las clases del paquete *configurations*, donde se define la configuración general para la ejecución del algoritmo.

En la Figura 11 se observa la existencia de una relación de uno a muchos entre las clases *AGeneticAlgorithm* y *GAMDPopulation*. Esto es un indicativo de que la clase *AGeneticAlgorithm* podrá definir una lista con N instancias de la clase *GAMDPopulation* para poder ejecutar el algoritmo sobre diferentes poblaciones. Se puede observar el mismo tipo de relación entre las clases *GAMDPopulation* y *AMDHead*. Esto se debe a que la clase *GAMDPopulation* se define sobre una lista de N instancias de *AMDHead*, que representaría la población actual de cromosomas que se está analizando.

Esta jerarquía de clases permitirá implementar nuevos operadores evolutivos, DMs, funciones de calidad o filtros sin necesidad de modificar el código fuente, solo haciendo uso del API que aquí se define. En este sentido los códigos fuentes 1 y 2 en los anexos muestran los pasos necesarios para agregar un nuevo DM sin modificar el código fuente del API. Por último, el **Código fuente 3** presenta los pasos básicos para invocar el API desde otra aplicación. Esto permite incluir nuestros algoritmos en otras herramientas o *pipelines*.

6.5 Conclusiones parciales

En este capítulo se presentó el software AExOp-DCS, el cual permite la ejecución de los algoritmos AExOp-DCS y C-AExOp-DCS en diferentes ambientes. Este programa está integrado por un API de desarrollo y dos interfaces de usuario: una basada en líneas de comandos y una basada en interfaces

gráficas de escritorio. Esto permite que los algoritmos propuestos en esta investigación puedan ser ejecutados en otras aplicaciones. Las interfaces desarrolladas permiten la ejecución del programa en diferentes entornos como pueden ser las terminales sin soporte para *X-Windows* o clúster de alto desempeño. Finalmente, esta herramienta fue desarrollada en Java 8, lo que garantiza además que pueda ser ejecutada en distintos sistemas operativos siempre y cuando tengan una máquina virtual de Java.

Capítulo 7. Conclusiones y trabajo futuro

7.1 Conclusiones

El análisis estadístico basado en estimación Bayesiana confirmó que los modelos creados a partir de los conjuntos de DMs devueltos por ambos algoritmos propuestos presentan mayor probabilidad de obtener mejor rendimiento que los modelos creados a partir de conjuntos de DMs sin optimizar.

El análisis estadístico basado en estimación Bayesiana confirmó que los modelos creados a partir de los conjuntos de DMs devueltos por el algoritmo cooperativo presenta una mayor probabilidad de obtener mejor rendimiento que los modelos creados a partir de conjuntos de DMs sin optimizar.

Los modelos construidos para la predicción de DILI construidos partiendo de subconjuntos de DMs retornados por la propuesta cooperativa alcanzaron rendimiento similar o superior a los mejores modelos reportados en la literatura.

En conclusión, los algoritmos propuestos resultan más adecuados para la modelación QSAR que el enfoque actualmente aplicado para obtener conjuntos de Descriptores Moleculares (DMs), incluso cuando la complejidad y diversidad de los compuestos crece.

7.2 Trabajo futuro

Trabajar en funciones de búsqueda de subconjunto que mantengan la calidad de la función actualmente en uso (selección de características basado en la correlación) pero que tenga mejor rendimiento en cuanto tiempo de ejecución.

Trabajar en una función de calidad de subconjunto basada en la integral de Choquet.

Actualmente la función de mérito de subconjunto CFS está basada en la correlación respecto a la clase, evaluar extender dicha función para que utilice la función de calidad multicriterio basada en la integral de Choquet definida en este trabajo.

Extender ambos algoritmos para que puedan utilizar otros DMs.

Definir ambos algoritmos para macromoléculas.

Aplicar ambos algoritmos en la búsqueda de subconjuntos de DMs para la predicción de otras actividades químico-físicas, para así validar la utilidad y efectividad de los subconjuntos devueltos en otros escenarios.

Validar el funcionamiento de los algoritmos cooperativos y no cooperativos sobre un conjunto de casos de prueba didáctico para caracterizar mejor el desempeño de los algoritmos propuestos.

Literatura citada

- Abdelrahman, M. A., Salama, I., Gomaa, M. S., Elaasser, M. M., Abdel-Aziz, M. M., & Soliman, D. H. (2017). Design, synthesis and 2D QSAR study of novel pyridine and quinolone hydrazone derivatives as potential antimicrobial and antitubercular agents. *European Journal of Medicinal Chemistry*, *138*, 698–714. <https://doi.org/10.1016/j.ejmech.2017.07.004>
- Aguilera-Mendoza, L., Marrero-Ponce, Y., Tellez-Ibarra, R., Llorente-Quesada, M. T., Salgado, J., Barigye, S. J., & Liu, J. (2015). Overlap and diversity in antimicrobial peptide databases: compiling a non-redundant set of sequences. *Bioinformatics*, *31*(15), 2553–2559. <https://doi.org/10.1093/bioinformatics/btv180>
- Akrami, A., & Niazi, A. (2017). Application of MIA for a QSAR Study of Inhibitory Activity of DHP Derivatives and Design of New Compounds Using WT and GA for Pixel Processing. *Polycyclic Aromatic Compounds*, *37*(5), 442–455. <https://doi.org/10.1080/10406638.2015.1129978>
- Al-Fakih, A. M., Algamal, Z. Y., Lee, M. H., Aziz, M., & Ali, H. T. M. (2019). QSAR classification model for diverse series of antifungal agents based on improved binary differential search algorithm. *SAR and QSAR in Environmental Research*, *30*(2), 131–143. <https://doi.org/10.1080/1062936X.2019.1568298>
- Ambure, P., Gajewicz-Skretna, A., Cordeiro, M. N. D. S., & Roy, K. (2019). New Workflow for QSAR Model Development from Small Data Sets: Small Dataset Curator and Small Dataset Modeler. Integration of Data Curation, Exhaustive Double Cross-Validation, and a Set of Optimal Model Selection Techniques. *Journal of Chemical Information and Modeling*, *59*(10), 4070–4076. <https://doi.org/10.1021/acs.jcim.9b00476>
- Ambure, P., Halder, A. K., González Díaz, H., & Cordeiro, M. N. D. S. (2019). QSAR-Co: An Open Source Software for Developing Robust Multitasking or Multitarget Classification-Based QSAR Models. *Journal of Chemical Information and Modeling*, *59*(6), 2538–2544. <https://doi.org/10.1021/acs.jcim.9b00295>
- Ansari, S. M., & Palmer, D. S. (2018). Comparative Molecular Field Analysis Using Molecular Integral Equation Theory. *Journal of Chemical Information and Modeling*, *58*(6), 1253–1265. <https://doi.org/10.1021/acs.jcim.7b00600>
- Antanasijević, D., Antanasijević, J., Trišović, N., Ušćumlić, G., & Pocaajt, V. (2017). From Classification to Regression Multitasking QSAR Modeling Using a Novel Modular Neural Network: Simultaneous Prediction of Anticonvulsant Activity and Neurotoxicity of Succinimides. *Molecular Pharmaceutics*, *14*(12), 4476–4484. <https://doi.org/10.1021/acs.molpharmaceut.7b00582>
- APACHE. *Apache Maven*. (2023, agosto 2). <https://maven.apache.org/>
- Bai, Y.-B., Gao, Y.-Q., Nie, X.-D., Tuong, T.-M.-L., Li, D., & Gao, J.-M. (2019). Antifungal Activity of Griseofulvin Derivatives against Phytopathogenic Fungi in Vitro and in Vivo and Three-Dimensional Quantitative Structure–Activity Relationship Analysis. *Journal of Agricultural and Food Chemistry*, *67*(22), 6125–6132. <https://doi.org/10.1021/acs.jafc.9b00606>
- Ballabio, D., Grisoni, F., Consonni, V., & Todeschini, R. (2019). Integrated QSAR Models to Predict Acute Oral Systemic Toxicity. *Molecular Informatics*, *38*(8–9), 1800124. <https://doi.org/10.1002/minf.201800124>

- Banerjee, P., Eckert, A. O., Schrey, A. K., & Preissner, R. (2018). ProTox-II: A webserver for the prediction of toxicity of chemicals. *Nucleic Acids Research*, 46(W1), W257–W263. <https://doi.org/10.1093/nar/gky318>
- Barigye, S., Marrero-Ponce, Y., Santiago, O., Lopez, Y., Perez-Gimenez, F., & Torrens, F. (2013). Shannon's, Mutual, Conditional and Joint Entropy Information Indices: Generalization of Global Indices Defined from Local Vertex Invariants. *Current Computer Aided-Drug Design*, 9(2), 164–183. <https://doi.org/10.2174/1573409911309020003>
- Beliakov, G., Bustince Sola, H., & Calvo Sánchez, T. (2016). *Fuzzy Integrals* (Vol. 329, pp. 145–181). https://doi.org/10.1007/978-3-319-24753-3_4
- Bellera, C. L., & Talevi, A. (2019). Quantitative structure–activity relationship models for compounds with anticonvulsant activity. *Expert Opinion on Drug Discovery*, 14(7), 653–665. <https://doi.org/10.1080/17460441.2019.1613368>
- Bemis, G. W., & Murcko, M. A. (1996). The properties of known drugs. 1. Molecular frameworks. *Journal of Medicinal Chemistry*, 39(15). <https://doi.org/10.1021/jm9602928>
- Benavoli, A., Corani, G., Demšar, J., & Zaffalon, M. (2017). Time for a Change: A Tutorial for Comparing Multiple Classifiers through Bayesian Analysis. *J. Mach. Learn. Res.*, 18(1), 2653–2688. <https://dl.acm.org/doi/10.5555/3122009.3176821>
- Benfenati, E. (Ed.). (2022). *In Silico Methods for Predicting Drug Toxicity* (Vol. 2425). Springer US. <https://doi.org/10.1007/978-1-0716-1960-5>
- Bolón-Canedo, V., & Alonso-Betanzos, A. (2019). Ensembles for feature selection: A review and future trends. *Information Fusion*, 52, 1–12. <https://doi.org/10.1016/j.inffus.2018.11.008>
- Bonachéra, F., & Horvath, D. (2008). Fuzzy Tricentric Pharmacophore Fingerprints. 2. Application of Topological Fuzzy Pharmacophore Triplets in Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling*, 48(2), 409–425. <https://doi.org/10.1021/ci7003237>
- Božić, A. R., Bjelogrić, S. K., Novaković, I. T., Filipović, N. R., Petrović, P. M., Marinković, A. D., Todorović, T. R., & Cvijetić, I. N. (2018). Antimicrobial Activity of Thiocarbohydrazones: Experimental Studies and Alignment-Independent 3D QSAR Models. *ChemistrySelect*, 3(7), 2215–2221. <https://doi.org/10.1002/slct.201702691>
- Brown, D. G., & Wobst, H. J. (2021). A Decade of FDA-Approved Drugs (2010–2019): Trends and Future Directions. *Journal of Medicinal Chemistry*, 64(5), 2312–2338. <https://doi.org/10.1021/acs.jmedchem.0c01516>
- Burke, E. K., & Kendall, G. (Eds.). (2014). *Search Methodologies: Introductory tutorials in optimization and decision support techniques*. (2nd ed.). Springer US. <https://doi.org/10.1007/978-1-4614-6940-7>
- Burley, S. K., Berman, H. M., Bhikadiya, C., Bi, C., Chen, L., Costanzo, L. Di, Christie, C., Duarte, J. M., Dutta, S., Feng, Z., Ghosh, S., Goodsell, D. S., Green, R. K., Guranovic, V., Guzenko, D., Hudson, B. P., Liang, Y., Lowe, R., Peisach, E., ... Ioannidis, Y. E. (2019). Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Research*, 47(D1), D520–D528. <https://doi.org/10.1093/nar/gky949>

- ChemAxon. (2023, agosto 2). <https://chemaxon.com/>
- Chen, S., Zhang, P., Liu, X., Qin, C., Tao, L., Zhang, C., Yang, S. Y., Chen, Y. Z., & Chui, W. K. (2016). Towards cheminformatics-based estimation of drug therapeutic index: Predicting the protective index of anticonvulsants using a new quantitative structure-index relationship approach. *Journal of Molecular Graphics and Modelling*, *67*, 102–110. <https://doi.org/10.1016/j.jmgm.2016.05.006>
- Choquet, G. (1954). Theory of capacities. *Annales de l'institut Fourier*, *5*, 131–295. <https://doi.org/10.5802/aif.53>
- Consonni, V., Ballabio, D., & Todeschini, R. (2010). Evaluation of model predictive ability by external validation techniques. *Journal of Chemometrics*, *24*(3–4), 194–201. <https://doi.org/10.1002/cem.1290>
- Consonni, V., & Todeschini, R. (2010). *Molecular Descriptors* (pp. 29–102). https://doi.org/10.1007/978-1-4020-9783-6_3
- da Silva, D. L., Silva Terra, B., Ribeiro Lage, M., Lúcia Tasca Góis Ruiz, A., Capeletti da Silva, C., Ernesto de Carvalho, J., Walkimar de Mesquita Carneiro, J., Terra Martins, F., Antonio Fernandes, S., & de Fátima, Â. (2015). Xanthenones: calixarenes-catalyzed syntheses, anticancer activity and QSAR studies. *Organic & Biomolecular Chemistry*, *13*(11), 3280–3287. <https://doi.org/10.1039/C4OB02611J>
- Deep, A., Narasimhan, B., Lim, S. M., Ramasamy, K., Mishra, R. K., & Mani, V. (2016). 4-Thiazolidinone derivatives: synthesis, antimicrobial, anticancer evaluation and QSAR studies. *RSC Advances*, *6*(111), 109485–109494. <https://doi.org/10.1039/C6RA23006G>
- Devinyak, O., Havrylyuk, D., & Lesyk, R. (2014). 3D-MoRSE descriptors explained. *Journal of Molecular Graphics and Modelling*, *54*, 194–203. <https://doi.org/10.1016/j.jmgm.2014.10.006>
- Dhingra, R., Malhotra, M., Sharma, V., Bhardwaj, T. R., & Dhingra, N. (2019). Exploration of Novel 5 α -Reductase Inhibitors for Benign Prostatic Hyperplasia by 2D/3D QSAR, Cytotoxicity Pre-ADME and Docking Studies. *Current Topics in Medicinal Chemistry*, *18*(32), 2816–2834. <https://doi.org/10.2174/1568026619666190119145959>
- Dolezal, R., Soukup, O., Malinak, D., Savedra, R. M. L., Marek, J., Dolezalova, M., Pasdiorova, M., Salajkova, S., Korabecny, J., Honegr, J., Ramalho, T. C., & Kuca, K. (2016). Towards understanding the mechanism of action of antibacterial N-alkyl-3-hydroxypyridinium salts: Biological activities, molecular modeling and QSAR studies. *European Journal of Medicinal Chemistry*, *121*, 699–711. <https://doi.org/10.1016/j.ejmech.2016.05.058>
- Dong, J., Wang, N. N., Yao, Z. J., Zhang, L., Cheng, Y., Ouyang, D., Lu, A. P., & Cao, D. S. (2018). Admetlab: A platform for systematic ADMET evaluation based on a comprehensively collected ADMET database. *Journal of Cheminformatics*, *10*(1). <https://doi.org/10.1186/s13321-018-0283-x>
- El-Zahabi, H. S. A., Khalifa, M. M. A., Gado, Y. M. H., Farrag, A. M., Elaasser, M. M., Safwat, N. A., AbdelRaouf, R. R., & Arafa, R. K. (2019). New thiobarbituric acid scaffold-based small molecules: Synthesis, cytotoxicity, 2D-QSAR, pharmacophore modelling and in-silico ADME screening. *European Journal of Pharmaceutical Sciences*, *130*, 124–136. <https://doi.org/10.1016/j.ejps.2019.01.023>

- Eshelman, L. J. (1991). *The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination* (pp. 265–283). <https://doi.org/10.1016/B978-0-08-050684-5.50020-3>
- Faidallah, H. M., Girgis, A. S., Tiwari, A. D., Honkanadavar, H. H., Thomas, S. J., Samir, A., Kalmouch, A., Alamry, K. A., Khan, K. A., Ibrahim, T. S., AL-Mahmoudy, A. M. M., Asiri, A. M., & Panda, S. S. (2018). Synthesis, antibacterial properties and 2D-QSAR studies of quinolone-triazole conjugates. *European Journal of Medicinal Chemistry*, *143*, 1524–1534. <https://doi.org/10.1016/j.ejmech.2017.10.042>
- García-Jacas, C. R., Aguilera-Mendoza, L., González-Pérez, R., Marrero-Ponce, Y., Acevedo-Martínez, L., Barigye, S. J., & Avdeenko, T. (2015). Multi-Server Approach for High-Throughput Molecular Descriptors Calculation based on Multi-Linear Algebraic Maps. *Molecular Informatics*, *34*(1), 60–69. <https://doi.org/10.1002/minf.201400086>
- García-Jacas, C. R., Cabrera-Leyva, L., Marrero-Ponce, Y., Suárez-Lezcano, J., Cortés-Guzmán, F., & García-González, L. A. (2018). GOWAWA Aggregation Operator-based Global Molecular Characterizations: Weighting Atom/bond Contributions (LOVIs/LOEIs) According to their Influence in the Molecular Encoding. *Molecular Informatics*, *37*(12). <https://doi.org/10.1002/minf.201800039>
- García-Jacas, C. R., Cabrera-Leyva, L., Marrero-Ponce, Y., Suárez-Lezcano, J., Cortés-Guzmán, F., Pupo-Meriño, M., & Vivas-Reyes, R. (2018). Choquet integral-based fuzzy molecular characterizations: when global definitions are computed from the dependency among atom/bond contributions (LOVIs/LOEIs). *Journal of Cheminformatics*, *10*(1), 51. <https://doi.org/10.1186/s13321-018-0306-7>
- García-Jacas, C. R., Marrero-Ponce, Y., Acevedo-Martínez, L., Barigye, S. J., Valdés-Martín, J. R., & Contreras-Torres, E. (2014). QuBiLS-MIDAS: A parallel free-software for molecular descriptors computation based on multilinear algebraic maps. *Journal of Computational Chemistry*, *35*(18), 1395–1409. <https://doi.org/10.1002/jcc.23640>
- García-Jacas, C. R., Marrero-Ponce, Y., Barigye, S. J., Hernández-Ortega, T., Cabrera-Leyva, L., & Fernández-Castillo, A. (2016). N -tuple topological/geometric cutoffs for 3D N -linear algebraic molecular codifications: variability, linear independence and QSAR analysis. *SAR and QSAR in Environmental Research*, *27*(12), 949–975. <https://doi.org/10.1080/1062936X.2016.1231714>
- García-Jacas, C. R., Marrero-Ponce, Y., Brizuela, C. A., Suárez-Lezcano, J., & Martínez-Ríos, F. (2020). Smoothed Spherical Truncation based on Fuzzy Membership Functions: Application to the Molecular Encoding. *Journal of Computational Chemistry*, *41*(3), 203–217. <https://doi.org/10.1002/jcc.26089>
- García-Jacas, C. R., Marrero-Ponce, Y., Cortés-Guzmán, F., Suárez-Lezcano, J., Martínez-Ríos, F. O., García-González, L. A., Pupo-Meriño, M., & Martínez-Mayorga, K. (2019). Enhancing Acute Oral Toxicity Predictions by using Consensus Modeling and Algebraic Form-Based 0D-to-2D Molecular Encodes. *Chemical Research in Toxicology*, *32*(6), 1178–1192. <https://doi.org/10.1021/acs.chemrestox.9b00011>
- García-Jacas, C. R., Marrero-Ponce, Y., Vivas-Reyes, R., Suárez-Lezcano, J., Martínez-Ríos, F., Terán, J. E., & Aguilera-Mendoza, L. (2020). Distributed and multicore QuBiLS-MIDAS software v2.0: Computing chiral, fuzzy, weighted and truncated geometrical molecular descriptors based on

- tensor algebra. *Journal of Computational Chemistry*, 41(12), 1209–1227. <https://doi.org/10.1002/jcc.26167>
- García-Pereira, I., Zanni, R., Galvez-Llompart, M., Galvez, J., & García-Domenech, R. (2019). DesMol2, an Effective Tool for the Construction of Molecular Libraries and Its Application to QSAR Using Molecular Topology. *Molecules*, 24(4), 736. <https://doi.org/10.3390/molecules24040736>
- Garro Martinez, J. C., Vega-Hissi, E. G., Andrada, M. F., & Estrada, M. R. (2015). QSAR and 3D-QSAR studies applied to compounds with anticonvulsant activity. *Expert Opinion on Drug Discovery*, 10(1), 37–51. <https://doi.org/10.1517/17460441.2015.968123>
- Gasteiger, J. (Ed.). (2003). *Handbook of Chemoinformatics*. Wiley. <https://doi.org/10.1002/9783527618279>
- 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>
- Ghanem, O. Ben, Shah, S. N., Lévêque, J.-M., Mutalib, M. I. A., El-Harbawi, M., Khan, A. S., Alnarabiji, M. S., Al-Absi, H. R. H., & Ullah, Z. (2018). Study of the antimicrobial activity of cyclic cation-based ionic liquids via experimental and group contribution QSAR model. *Chemosphere*, 195, 21–28. <https://doi.org/10.1016/j.chemosphere.2017.12.018>
- Gholivand, K., EbrahimiValmoozi, A. A., Gholami, A., Dusek, M., Eigner, V., & Abolghasemi, S. (2016). Synthesis, characterization, crystal structures, QSAR study and antibacterial activities of organotin bisphosphoramidates. *Journal of Organometallic Chemistry*, 806, 33–44. <https://doi.org/10.1016/j.jorganchem.2015.09.030>
- Gini, G., Zanoli, F., Gamba, A., Raitano, G., & Benfenati, E. (2019). Could deep learning in neural networks improve the QSAR models? *SAR and QSAR in Environmental Research*, 30(9), 617–642. <https://doi.org/10.1080/1062936X.2019.1650827>
- Godden, J. W., Stahura, F. L., & Bajorath, J. (2000). Variability of Molecular Descriptors in Compound Databases Revealed by Shannon Entropy Calculations. *Journal of Chemical Information and Computer Sciences*, 40(3), 796–800. <https://doi.org/10.1021/ci000321u>
- Gökçe, S., & Saçan, M. T. (2019). Assessments of Algal Toxicity and PBT Behaviour of Pesticides with No Eco-toxicological Data: Predictive Ability of QSA(T)R Models. *Molecular Informatics*, 38(8–9), 1800137. <https://doi.org/10.1002/minf.201800137>
- Golbraikh, A., & Tropsha, A. (2002). Beware of q²! *Journal of Molecular Graphics and Modelling*, 20(4), 269–276. [https://doi.org/10.1016/S1093-3263\(01\)00123-1](https://doi.org/10.1016/S1093-3263(01)00123-1)
- Golbraikh, A., Wang, X. S., Zhu, H., & Tropsha, A. (2017). Predictive QSAR Modeling: Methods and Applications in Drug Discovery and Chemical Risk Assessment. En *Handbook of Computational Chemistry* (pp. 2303–2340). Springer International Publishing. https://doi.org/10.1007/978-3-319-27282-5_37
- Grabisch, M. (1995). Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 69(3), 279–298. [https://doi.org/10.1016/0165-0114\(94\)00174-6](https://doi.org/10.1016/0165-0114(94)00174-6)

- Grabisch, M., & Labreuche, C. (2005). Fuzzy Measures and Integrals in MCDA. In *International series in management science/operations research*, 563–604. https://doi.org/10.1007/0-387-23081-5_14
- Gramatica, P. (2006). WHIM Descriptors of Shape. *QSAR & Combinatorial Science*, 25(4), 327–332. <https://doi.org/10.1002/qsar.200510159>
- Gramatica, P. (2007). Principles of QSAR models validation: internal and external. *QSAR & Combinatorial Science*, 26(5), 694–701. <https://doi.org/10.1002/qsar.200610151>
- Gramatica, P., & Sangion, A. (2016). A Historical Excursus on the Statistical Validation Parameters for QSAR Models: A Clarification Concerning Metrics and Terminology. *Journal of Chemical Information and Modeling*, 56(6), 1127–1131. <https://doi.org/10.1021/acs.jcim.6b00088>
- Grimm, D. (2019). EPA plan to end animal testing splits scientists. *Science*, 365(6459), 1231–1231. <https://doi.org/10.1126/science.365.6459.1231>
- Grisoni, F., Ballabio, D., Todeschini, R., & Consonni, V. (2018). *Molecular Descriptors for Structure–Activity Applications: A Hands-On Approach* (pp. 3–53). https://doi.org/10.1007/978-1-4939-7899-1_1
- Gupta, M. K., Gupta, S., & Rawal, R. K. (2016). Impact of Artificial Neural Networks in QSAR and Computational Modeling. En *Artificial Neural Network for Drug Design, Delivery and Disposition* (pp. 153–179). Elsevier. <https://doi.org/10.1016/B978-0-12-801559-9.00008-9>
- Hall, M. A. (2000). Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning. *Proceedings of the Seventeenth International Conference on Machine Learning, April*, 359–366. <https://dl.acm.org/doi/10.5555/645529.657793>
- He, Y., Liew, C. Y., Sharma, N., Woo, S. K., Chau, Y. T., & Yap, C. W. (2013). PaDEL-DDPredictor: Open-source software for PD-PK-T prediction. *Journal of Computational Chemistry*, 34(7). <https://doi.org/10.1002/jcc.23173>
- Hemmer, M. C., Steinhauer, V., & Gasteiger, J. (1999). Deriving the 3D structure of organic molecules from their infrared spectra. *Vibrational Spectroscopy*, 19(1), 151–164. [https://doi.org/10.1016/S0924-2031\(99\)00014-4](https://doi.org/10.1016/S0924-2031(99)00014-4)
- Hinselman, G., Rosenbaum, L., Jahn, A., Fechner, N., & Zell, A. (2011). jCompoundMapper: An open source Java library and command-line tool for chemical fingerprints. *Journal of Cheminformatics*, 3(1), 3. <https://doi.org/10.1186/1758-2946-3-3>
- Hodyna, D., Kovalishyn, V., Rogalsky, S., Blagodatnyi, V., Petko, K., & Metelytsia, L. (2016). Antibacterial Activity of Imidazolium-Based Ionic Liquids Investigated by QSAR Modeling and Experimental Studies. *Chemical Biology & Drug Design*, 88(3), 422–433. <https://doi.org/10.1111/cbdd.12770>
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press. <https://doi.org/10.7551/mitpress/1090.001.0001>
- Hsiang-Chuan Liu and Der-Bang Wu and Yu-Du Jheng and Tian-Wei Sheu. (2009). Theory of multivalent delta-fuzzy measures and its application. *WSEAS Transactions on Information Science and Applications archive*, 6, 1061–1070. <https://dl.acm.org/doi/10.5555/1639438.1639455>

- Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., & Coleman, R. G. (2012). ZINC: A Free Tool to Discover Chemistry for Biology. *Journal of Chemical Information and Modeling*, 52(7), 1757–1768. <https://doi.org/10.1021/ci3001277>
- JETBRAINS. *IntelliJ IDEA*. (2023, agosto 2). <https://www.jetbrains.com/idea/>.
- Kaitoh, K., Kotera, M., & Funatsu, K. (2019). Novel Electrotopological Atomic Descriptors for the Prediction of Xenobiotic Cytochrome P450 Reactions. *Molecular Informatics*, 38(10), 1900010. <https://doi.org/10.1002/minf.201900010>
- Kallner, A. (2018). Formulas. En *Laboratory Statistics* (pp. 1–140). Elsevier. <https://doi.org/10.1016/B978-0-12-814348-3.00001-0>
- Kang, M. G., & Kang, N. S. (2021). Predictive model for drug-induced liver injury using deep neural networks based on substructure space. *Molecules*, 26(24). <https://doi.org/10.3390/molecules26247548>
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5), 8091–8126. <https://doi.org/10.1007/s11042-020-10139-6>
- Kawczak, P., Bober, L., & Bączek, T. (2018). Application of QSAR Analysis and Different Quantum Chemical Calculation Methods in Activity Evaluation of Selected Fluoroquinolones. *Combinatorial Chemistry & High Throughput Screening*, 21(7), 468–475. <https://doi.org/10.2174/1386207321666180827105856>
- Keying Ye, Raymond Myers, Roland Walpole, & Sharon Myers. (2012). *Probabilidad y estadística para ingeniería y ciencias* (9a ed.). Pearson Educación de México.
- Khan, K., Benfenati, E., & Roy, K. (2019). Consensus QSAR modeling of toxicity of pharmaceuticals to different aquatic organisms: Ranking and prioritization of the DrugBank database compounds. *Ecotoxicology and Environmental Safety*, 168, 287–297. <https://doi.org/10.1016/j.ecoenv.2018.10.060>
- Kim, J. H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, 53(11). <https://doi.org/10.1016/j.csda.2009.04.009>
- Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J., & Bolton, E. E. (2019). PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1), D1102–D1109. <https://doi.org/10.1093/nar/gky1033>
- Kira, K., & Rendell, L. A. (1992). A Practical Approach to Feature Selection. En *Machine Learning Proceedings 1992* (pp. 249–256). Elsevier. <https://doi.org/10.1016/B978-1-55860-247-2.50037-1>
- Klamt, A., Thormann, M., Wichmann, K., & Tosco, P. (2012). COSMO sar3D : Molecular Field Analysis Based on Local COSMO σ -Profiles. *Journal of Chemical Information and Modeling*, 52(8), 2157–2164. <https://doi.org/10.1021/ci300231t>

- Klüver, N., Bittermann, K., & Escher, B. I. (2019). QSAR for baseline toxicity and classification of specific modes of action of ionizable organic chemicals in the zebrafish embryo toxicity test. *Aquatic Toxicology*, 207, 110–119. <https://doi.org/10.1016/j.aquatox.2018.12.003>
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2), 273–324. [https://doi.org/10.1016/s0004-3702\(97\)00043-x](https://doi.org/10.1016/s0004-3702(97)00043-x)
- Krishnan, A. R., Aqilah, S. N., Kasim, M. M., Nazri, E. M., & Char, A. K. (2017). A revised procedure to identify λ 0-measure values for applying Choquet integral in solving multi-attribute decision problems. *OPSEARCH*, 54(3), 637–650. <https://doi.org/10.1007/s12597-017-0297-6>
- Krzyzanowski, A., Pahl, A., Grigalunas, M., & Waldmann, H. (2023). Spacial Score—A Comprehensive Topological Indicator for Small-Molecule Complexity. *Journal of Medicinal Chemistry*. <https://doi.org/10.1021/acs.jmedchem.3c00689>
- Kuz'min, V. E., Ognichenko, L. N., Sizochenko, N., Chapkin, V. A., Stelmakh, S. I., Shyrykalova, A. O., & Leszczynski, J. (2019). Combining Features of Metal Oxide Nanoparticles. *International Journal of Quantitative Structure-Property Relationships*, 4(1), 28–40. <https://doi.org/10.4018/IJQSPR.2019010103>
- Lavado, G. J., Baderna, D., Gadaleta, D., Ulte, M., Roy, K., & Benfenati, E. (2021). Ecotoxicological QSAR modeling of the acute toxicity of organic compounds to the freshwater crustacean *Thamnocephalus platyurus*. *Chemosphere*, 280, 130652. <https://doi.org/10.1016/j.chemosphere.2021.130652>
- Lavinias, Y., Aranha, C., Sakurai, T., & Ladeira, M. (2018). Experimental Analysis of the Tournament Size on Genetic Algorithms. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3647–3653. <https://doi.org/10.1109/SMC.2018.00617>
- Lee, A., Mercader, A. G., Duchowicz, P. R., Castro, E. A., & Pomilio, A. B. (2012). QSAR study of the DPPH radical scavenging activity of di(hetero)arylamines derivatives of benzo[b]thiophenes, halophenols and caffeic acid analogues. *Chemometrics and Intelligent Laboratory Systems*, 116, 33–40. <https://doi.org/10.1016/j.chemolab.2012.03.016>
- Leemans, E., Mahasenani, K. V., Kumarasiri, M., Spink, E., Ding, D., O'Daniel, P. I., Boudreau, M. A., Lastochkin, E., Testero, S. A., Yamaguchi, T., Lee, M., Heseck, D., Fisher, J. F., Chang, M., & Mobashery, S. (2016). Three-dimensional QSAR analysis and design of new 1,2,4-oxadiazole antibacterials. *Bioorganic & Medicinal Chemistry Letters*, 26(3), 1011–1015. <https://doi.org/10.1016/j.bmcl.2015.12.041>
- Leslie, A. B., Simpson, C., & Mander, L. (2021). Reproductive innovations and pulsed rise in plant complexity. *Science*, 373(6561), 1368–1372. <https://doi.org/10.1126/science.abi6984>
- Li, T., Tong, W., Roberts, R., Liu, Z., & Thakkar, S. (2021). DeepDILI: Deep Learning-Powered Drug-Induced Liver Injury Prediction Using Model-Level Representation. *Chemical Research in Toxicology*, 34(2). <https://doi.org/10.1021/acs.chemrestox.0c00374>
- Li, X., Chen, Y., Song, X., Zhang, Y., Li, H., & Zhao, Y. (2018). The development and application of *in silico* models for drug induced liver injury. *RSC Advances*, 8(15), 8101–8111. <https://doi.org/10.1039/C7RA12957B>

- Liew, C. Y., Lim, Y. C., & Yap, C. W. (2011). Mixed learning algorithms and features ensemble in hepatotoxicity prediction. *Journal of Computer-Aided Molecular Design*, 25(9), 855–871. <https://doi.org/10.1007/s10822-011-9468-3>
- Lino, C. I., Gonçalves de Souza, I., Borelli, B. M., Silvério Matos, T. T., Santos Teixeira, I. N., Ramos, J. P., Maria de Souza Fagundes, E., de Oliveira Fernandes, P., Maltarollo, V. G., Johann, S., & de Oliveira, R. B. (2018). Synthesis, molecular modeling studies and evaluation of antifungal activity of a novel series of thiazole derivatives. *European Journal of Medicinal Chemistry*, 151, 248–260. <https://doi.org/10.1016/j.ejmech.2018.03.083>
- Liu, H.-C., Jheng, Y.-D., Lin, W.-C., & Chen, G.-S. (2007). A Novel Fuzzy Measure and its Choquet Integral Regression Model. *2007 International Conference on Machine Learning and Cybernetics*, 1394–1398. <https://doi.org/10.1109/ICMLC.2007.4370362>
- Liu, T., Yan, F., Jia, Q., & Wang, Q. (2020). Norm index-based QSAR models for acute toxicity of organic compounds toward zebrafish embryo. *Ecotoxicology and Environmental Safety*, 203, 110946. <https://doi.org/10.1016/j.ecoenv.2020.110946>
- Loupe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding Variable Importances in Forests of Randomized Trees. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, 431–439. <https://dl.acm.org/doi/10.5555/2999611.2999660>
- Lu, J., Peng, J., Wang, J., Shen, Q., Bi, Y., Gong, L., Zheng, M., Luo, X., Zhu, W., Jiang, H., & Chen, K. (2014). Estimation of acute oral toxicity in rat using local lazy learning. *Journal of Cheminformatics*, 6(1), 26. <https://doi.org/10.1186/1758-2946-6-26>
- Lu, P., Bevan, D. R., Leber, A., Hontecillas, R., Tubau-Juni, N., & Bassaganya-Riera, J. (2018). Computer-Aided Drug Discovery. En *Accelerated Path to Cures* (pp. 7–24). Springer International Publishing. https://doi.org/10.1007/978-3-319-73238-1_2
- Lust, T. (2015). *Choquet Integral Versus Weighted Sum in Multicriteria Decision Contexts* (pp. 288–304). https://doi.org/10.1007/978-3-319-23114-3_18
- Majumdar, S., & Basak, S. C. (2018). Beware of External Validation! - A Comparative Study of Several Validation Techniques used in QSAR Modelling. *Current Computer-Aided Drug Design*, 14(4), 284–291. <https://doi.org/10.2174/1573409914666180426144304>
- Manchester, J., & Czermiński, R. (2008). SAMFA: Simplifying Molecular Description for 3D-QSAR. *Journal of Chemical Information and Modeling*, 48(6), 1167–1173. <https://doi.org/10.1021/ci800009u>
- Marichal, J.-L. (2000). On Sugeno integral as an aggregation function. *Fuzzy Sets and Systems*, 114(3), 347–365. [https://doi.org/10.1016/S0165-0114\(98\)00116-X](https://doi.org/10.1016/S0165-0114(98)00116-X)
- Martínez, M. J., Dussaut, J. S., & Ponzoni, I. (2018). Biclustering as Strategy for Improving Feature Selection in Consensus QSAR Modeling. *Electronic Notes in Discrete Mathematics*, 69, 117–124. <https://doi.org/10.1016/j.endm.2018.07.016>
- Martínez-Santiago, O., Marrero-Ponce, Y., Vivas-Reyes, R., Rivera-Borroto, O. M., Hurtado, E., Treto-Suarez, M. A., Ramos, Y., Vergara-Murillo, F., Orozco-Ugarriza, M. E., & Martínez-López, Y. (2017). Exploring the QSAR's predictive truthfulness of the novel N -tuple discrete derivative

- indices on benchmark datasets. *SAR and QSAR in Environmental Research*, 28(5), 367–389. <https://doi.org/10.1080/1062936X.2017.1326403>
- Mauri, A. (2020). *alvaDesc: A Tool to Calculate and Analyze Molecular Descriptors and Fingerprints* (pp. 801–820). https://doi.org/10.1007/978-1-0716-0150-1_32
- Mauri, A., Consonni, V., Pavan, M., & Todeschini, R. (2006). DRAGON software: An easy approach to molecular descriptor calculations. *Match*, 56(2), 237–248. https://match.pmf.kg.ac.rs/electronic_versions/Match56/n2/match56n2_237-248.pdf
- Mauri, A., Consonni, V., & Todeschini, R. (2016). Molecular Descriptors. En *Handbook of Computational Chemistry* (pp. 1–29). Springer Netherlands https://doi.org/10.1007/978-94-007-6169-8_51-1
- Mohamed, M. A., & Weimin Xiao. (2003). Q-measures: an efficient extension of the sugeno λ -measure. *IEEE Transactions on Fuzzy Systems*, 11(3), 419–426. <https://doi.org/10.1109/TFUZZ.2003.812701>
- Mora, J. R., Marrero-Ponce, Y., García-Jacas, C. R., & Suarez Causado, A. (2020). Ensemble Models Based on QuBiLS-MAS Features and Shallow Learning for the Prediction of Drug-Induced Liver Toxicity: Improving Deep Learning and Traditional Approaches. *Chemical Research in Toxicology*, 33(7), 1855–1873. <https://doi.org/10.1021/acs.chemrestox.0c00030>
- Moriwaki, H., Tian, Y.-S., Kawashita, N., & Takagi, T. (2018). Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10(1), 4. <https://doi.org/10.1186/s13321-018-0258-y>
- Muegge, I., Bergner, A., & Kriegl, J. M. (2017). Computer-aided drug design at Boehringer Ingelheim. *Journal of Computer-Aided Molecular Design*, 31(3), 275–285. <https://doi.org/10.1007/s10822-016-9975-3>
- Muratov, E. N., Bajorath, J., Sheridan, R. P., Tetko, I. V., Filimonov, D., Poroikov, V., Oprea, T. I., Baskin, I. I., Varnek, A., Roitberg, A., Isayev, O., Curtalolo, S., Fourches, D., Cohen, Y., Aspuru-Guzik, A., Winkler, D. A., Agrafiotis, D., Cherkasov, A., & Tropsha, A. (2020). QSAR without borders. *Chemical Society Reviews*, 49(11), 3525–3564. <https://doi.org/10.1039/D0CS00098A>
- Murofushi, T., & Sugeno, M. (1991). A theory of fuzzy measures: Representations, the Choquet integral, and null sets. *Journal of Mathematical Analysis and Applications*, 159(2), 532–549. [https://doi.org/10.1016/0022-247X\(91\)90213-J](https://doi.org/10.1016/0022-247X(91)90213-J)
- Nagai, J., Imamura, M., Sakagami, H., & Uesawa, Y. (2019). QSAR Prediction Model to Search for Compounds with Selective Cytotoxicity Against Oral Cell Cancer. *Medicines*, 6(2), 45. <https://doi.org/10.3390/medicines6020045>
- Nath, A., De, P., & Roy, K. (2022). QSAR modelling of inhalation toxicity of diverse volatile organic molecules using no observed adverse effect concentration (NOAEC) as the endpoint. *Chemosphere*, 287, 131954. <https://doi.org/10.1016/j.chemosphere.2021.131954>
- Neves, B. J., Braga, R. C., Melo-Filho, C. C., Moreira-Filho, J. T., Muratov, E. N., & Andrade, C. H. (2018). QSAR-Based Virtual Screening: Advances and Applications in Drug Discovery. *Frontiers in Pharmacology*, 9. <https://doi.org/10.3389/fphar.2018.01275>

- Nguyen-Vo, T. H., Nguyen, L., Do, N., Le, P. H., Nguyen, T. N., Nguyen, B. P., & Le, L. (2020). Predicting Drug-Induced Liver Injury Using Convolutional Neural Network and Molecular Fingerprint-Embedded Features. *ACS Omega*, 5(39). <https://doi.org/10.1021/acsomega.0c03866>
- OECD. (2023, agosto 2). *OECD: Organisation for Economic Co-operation and Development*. <https://www.oecd.org/env/ehs/risk-assessment/validationofqsarmodels.htm>
- Oglic, D., Oatley, S. A., Macdonald, S. J. F., Mcinally, T., Garnett, R., Hirst, J. D., & Gärtner, T. (2018). Active Search for Computer-aided Drug Design. *Molecular Informatics*, 37(1–2), 1700130. <https://doi.org/10.1002/minf.201700130>
- Olague, G. (2016). *Evolutionary Computing*. In *Natural computing series* (pp. 69–140). https://doi.org/10.1007/978-3-662-43693-6_3
- ORACLE. *Oracle Java*. (2023, marzo 2). <https://www.oracle.com/mx/java/>.
- Panda, S. S., Liaqat, S., Girgis, A. S., Samir, A., Hall, C. D., & Katritzky, A. R. (2015). Novel antibacterial active quinolone–fluoroquinolone conjugates and 2D-QSAR studies. *Bioorganic & Medicinal Chemistry Letters*, 25(18), 3816–3821. <https://doi.org/10.1016/j.bmcl.2015.07.077>
- Papa, E., van der Wal, L., Arnot, J. A., & Gramatica, P. (2014). Metabolic biotransformation half-lives in fish: QSAR modeling and consensus analysis. *Science of The Total Environment*, 470–471, 1040–1046. <https://doi.org/10.1016/j.scitotenv.2013.10.068>
- Pes, B. (2020). Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains. *Neural Computing and Applications*, 32(10), 5951–5973. <https://doi.org/10.1007/s00521-019-04082-3>
- Pingaew, R., Prachayasittikul, V., Worachartcheewan, A., Nantasenamat, C., Prachayasittikul, S., Ruchirawat, S., & Prachayasittikul, V. (2015). Novel 1,4-naphthoquinone-based sulfonamides: Synthesis, QSAR, anticancer and antimalarial studies. *European Journal of Medicinal Chemistry*, 103, 446–459. <https://doi.org/10.1016/j.ejmech.2015.09.001>
- Pinto, V., Araújo, J., Silva, R., da Costa, G., Cruz, J., De A. Neto, M., Campos, J., Santos, C., Leite, F., & Junior, M. (2019). In Silico Study to Identify New Antituberculosis Molecules from Natural Sources by Hierarchical Virtual Screening and Molecular Dynamics Simulations. *Pharmaceuticals*, 12(1), 36. <https://doi.org/10.3390/ph12010036>
- Pires, D. E. V., Blundell, T. L., & Ascher, D. B. (2015). pkCSM: Predicting Small-Molecule Pharmacokinetic and Toxicity Properties Using Graph-Based Signatures. *Journal of Medicinal Chemistry*, 58(9), 4066–4072. <https://doi.org/10.1021/acs.jmedchem.5b00104>
- Pizzolitto, R. P., Jacquat, A. G., Usseglio, V. L., Achimón, F., Cuello, A. E., Zygadlo, J. A., & Dambolena, J. S. (2020). Quantitative-structure-activity relationship study to predict the antifungal activity of essential oils against *Fusarium verticillioides*. *Food Control*, 108, 106836. <https://doi.org/10.1016/j.foodcont.2019.106836>
- Potter, M. A., & Jong, K. A. De. (2000). Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation*, 8(1), 1–29. <https://doi.org/10.1162/106365600568086>

- Potter, M. A., & Jong, K. A. (1994). *A cooperative coevolutionary approach to function optimization* (pp. 249–257). https://doi.org/10.1007/3-540-58484-6_269
- Pourbasheer, E., Aalizadeh, R., Ganjali, M. R., & Norouzi, P. (2014). QSAR study of $\alpha 1\beta 4$ integrin inhibitors by GA-MLR and GA-SVM methods. *Structural Chemistry*, 25(1), 355–370. <https://doi.org/10.1007/s11224-013-0300-7>
- Pradhan, J., & Goyal, A. (2016). Synthesis, anticonvulsant activity and QSAR studies of some new pyrazolyl pyridines. *Medicinal Chemistry Research*, 25(8), 1639–1656. <https://doi.org/10.1007/s00044-016-1597-8>
- Rajathei, D. M., Parthasarathy, S., & Selvaraj, S. (2019). QSAR Analysis of Multimodal Antidepressants Vortioxetine Analogs Using Physicochemical Descriptors and MLR Modeling. *Current Computer-Aided Drug Design*, 15(4), 294–307. <https://doi.org/10.2174/1573409914666181011144810>
- RDKit: Open-Source Cheminformatics Software. (2023, agosto 2). <https://www.rdkit.org/>
- Rodriguez-Coayahuitl, L., Morales-Reyes, A., Escalante, H. J., & Coello Coello, C. A. (2020). *Cooperative Co-Evolutionary Genetic Programming for High Dimensional Problems* (pp. 48–62). https://doi.org/10.1007/978-3-030-58115-2_4
- Roy, K. (Ed.). (2020). *Ecotoxicological QSARs*. Springer US. <https://doi.org/10.1007/978-1-0716-0150-1>
- Roy, K., & Ambure, P. (2016). The “double cross-validation” software tool for MLR QSAR model development. *Chemometrics and Intelligent Laboratory Systems*, 159, 108–126. <https://doi.org/10.1016/j.chemolab.2016.10.009>
- Roy, K., Ambure, P., Kar, S., & Ojha, P. K. (2018). Is it possible to improve the quality of predictions from an “intelligent” use of multiple QSAR/QSPR/QSTR models? *Journal of Chemometrics*, 32(4), e2992. <https://doi.org/10.1002/cem.2992>
- Roy, K., Kar, S., & Das, R. N. (2015a). *Statistical Methods in QSAR/QSPR* (pp. 37–59). https://doi.org/10.1007/978-3-319-17281-1_2
- Roy, K., Kar, S., & Das, R. N. (2015b). *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*. Academic press. <https://doi.org/10.1016/C2014-0-00286-9>
- Roy, K., Mitra, I., Kar, S., Ojha, P. K., Das, R. N., & Kabir, H. (2012). Comparative Studies on Some Metrics for External Validation of QSPR Models. *Journal of Chemical Information and Modeling*, 52(2), 396–408. <https://doi.org/10.1021/ci200520g>
- Roy, P. P., Banjare, P., Verma, S., & Singh, J. (2019). Acute Rat and Mouse Oral Toxicity Determination of Anticholinesterase Inhibitor Carbamate Pesticides: A QSTR Approach. *Molecular Informatics*, 38(8–9), 1800151. <https://doi.org/10.1002/minf.201800151>
- Roy, P. P., Singh, J., & Ray, S. (2018). Exploring QSAR of Some Antitubercular Agents. *International Journal of Quantitative Structure-Property Relationships*, 3(1), 25–42. <https://doi.org/10.4018/IJQSPR.2018010102>
- Rücker, C., Rücker, G., & Meringer, M. (2007). Y-randomization and its variants in QSPR/QSAR. *Journal of Chemical Information and Modeling*, 47(6). <https://doi.org/10.1021/ci700157b>

- Sahasrabudhe, V., Zhu, T., Vaz, A., & Tse, S. (2015). Drug Metabolism and Drug Interactions: Potential Application to Antituberculosis Drugs. *Journal of Infectious Diseases*, 211(suppl 3), S107–S114. <https://doi.org/10.1093/infdis/jiv009>
- Saldívar-González, F. I., & Medina-Franco, J. L. (2020). Chemoinformatics approaches to assess chemical diversity and complexity of small molecules. En *Small Molecule Drug Discovery* (pp. 83–102). Elsevier. <https://doi.org/10.1016/B978-0-12-818349-6.00003-0>
- Sass, P. (Ed.). (2017). *Antibiotics* (Vol. 1520). Springer New York. <https://doi.org/10.1007/978-1-4939-6634-9>
- Schaub, J., Zander, J., Zielesny, A., & Steinbeck, C. (2022). Scaffold Generator: a Java library implementing molecular scaffold functionalities in the Chemistry Development Kit (CDK). *Journal of Cheminformatics*, 14(1), 79. <https://doi.org/10.1186/s13321-022-00656-x>
- Schyman, P., Liu, R., Desai, V., & Wallqvist, A. (2017). vNN Web Server for ADMET Predictions. *Frontiers in Pharmacology*, 8. <https://doi.org/10.3389/fphar.2017.00889>
- Shameera Ahamed, T. K., Rajan, V. K., & Muraleedharan, K. (2019). QSAR modeling of benzoquinone derivatives as 5-lipoxygenase inhibitors. *Food Science and Human Wellness*, 8(1), 53–62. <https://doi.org/10.1016/j.fshw.2019.02.001>
- Sharapova, A., Ol'khovich, M., Blokhina, S., & Perlovich, G. (2017). Physico-chemical characterization antituberculosis thioacetazone: Vapor pressure, solubility and lipophilicity. *The Journal of Chemical Thermodynamics*, 108, 18–25. <https://doi.org/10.1016/j.jct.2016.12.034>
- Sheffield, T. Y., & Judson, R. S. (2019). Ensemble QSAR Modeling to Predict Multispecies Fish Toxicity Lethal Concentrations and Points of Departure. *Environmental Science & Technology*, 53(21), 12793–12802. <https://doi.org/10.1021/acs.est.9b03957>
- Sławiński, J., Szafranski, K., Pogorzelska, A., Żołnowska, B., Kawiak, A., Macur, K., Belka, M., & Bączek, T. (2017). Novel 2-benzylthio-5-(1,3,4-oxadiazol-2-yl)benzenesulfonamides with anticancer activity: Synthesis, QSAR study, and metabolic stability. *European Journal of Medicinal Chemistry*, 132, 236–248. <https://doi.org/10.1016/j.ejmech.2017.03.039>
- Sun, H., Huang, R., Xia, M., Shahane, S., Southall, N., & Wang, Y. (2017). Prediction of hERG Liability - Using SVM Classification, Bootstrapping and Jackknifing. *Molecular Informatics*, 36(4), 1600126. <https://doi.org/10.1002/minf.201600126>
- Sun, L., Yang, H., Li, J., Wang, T., Li, W., Liu, G., & Tang, Y. (2018). In Silico Prediction of Compounds Binding to Human Plasma Proteins by QSAR Models. *ChemMedChem*, 13(6), 572–581. <https://doi.org/10.1002/cmdc.201700582>
- Sutherland, J. J., O'Brien, L. A., & Weaver, D. F. (2004). A Comparison of Methods for Modeling Quantitative Structure–Activity Relationships. *Journal of Medicinal Chemistry*, 47(22), 5541–5554. <https://doi.org/10.1021/jm0497141>
- Terán, J. E., Marrero-Ponce, Y., Contreras-Torres, E., García-Jacas, C. R., Vivas-Reyes, R., Terán, E., & Torres, F. J. (2019). Tensor Algebra-based Geometrical (3D) Biomacro-Molecular Descriptors for Protein Research: Theory, Applications and Comparison with other Methods. *Scientific Reports*, 9(1), 11391. <https://doi.org/10.1038/s41598-019-47858-2>

- The Validation of Alternative Test Methods* (pp. 307–314). (2019). Elsevier. <https://linkinghub.elsevier.com/retrieve/pii/B9780128136973000330>
- Timbrell, J. A. (2008). *Principles of Biochemical Toxicology*. CRC Press. <https://doi.org/10.3109/9781420007084>
- Todeschini, R., Ballabio, D., & Grisoni, F. (2016). Beware of Unreliable Q² ! A Comparative Study of Regression Metrics for Predictivity Assessment of QSAR Models. *Journal of Chemical Information and Modeling*, 56(10), 1905–1913. <https://doi.org/10.1021/acs.jcim.6b00277>
- Todeschini, R., Consonni, V., Ballabio, D., & Grisoni, F. (2020). Chemometrics for QSAR Modeling. En *Comprehensive Chemometrics* (pp. 599–634). Elsevier. <https://doi.org/10.1016/B978-0-12-409547-2.14703-1>
- Todeschini, R., Consonni, V., & Gramatica, P. (2009). Chemometrics in QSAR. En *Comprehensive Chemometrics* (pp. 129–172). Elsevier. <https://doi.org/10.1016/B978-044452701-1.00007-7>
- Toropova, A. P., Toropov, A. A., Beeg, M., Gobbi, M., & Salmona, M. (2017). Utilization of the Monte Carlo Method to Build up QSAR Models for Hemolysis and Cytotoxicity of Antimicrobial Peptides. *Current Drug Discovery Technologies*, 14(4). <https://doi.org/10.2174/1570163814666170525114128>
- Toropova, M. A., Veselinović, A. M., Veselinović, J. B., Stojanović, D. B., & Toropov, A. A. (2015). QSAR modeling of the antimicrobial activity of peptides as a mathematical function of a sequence of amino acids. *Computational Biology and Chemistry*, 59, 126–130. <https://doi.org/10.1016/j.compbiolchem.2015.09.009>
- Tosco, P., & Balle, T. (2012). A 3D-QSAR-Driven Approach to Binding Mode and Affinity Prediction. *Journal of Chemical Information and Modeling*, 52(2), 302–307. <https://doi.org/10.1021/ci200411s>
- Tropsha, A. (2010). Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics*, 29(6–7), 476–488. <https://doi.org/10.1002/minf.201000061>
- Tropsha, A., Gramatica, P., & Gombar, V. K. (2003). The importance of being earnest: Validation is the absolute essential for successful application and interpretation of QSPR models. *QSAR and Combinatorial Science*, 22(1). <https://doi.org/10.1002/qsar.200390007>
- Trush, M. M., Kovalishyn, V., Ocheretniuk, A. D., Kobzar, O. L., Kachaeva, M. V., Brovarets, V. S., & Metelytsia, L. O. (2019). QSAR Study of Some 1,3-Oxazolylphosphonium Derivatives as New Potent Anti-Candida Agents and Their Toxicity Evaluation. *Current Drug Discovery Technologies*, 16(2), 204–209. <https://doi.org/10.2174/1570163815666180418145422>
- Tseng, C.-H., Tung, C.-W., Wu, C.-H., Tzeng, C.-C., Chen, Y.-H., Hwang, T.-L., & Chen, Y.-L. (2017). Discovery of Indeno[1,2-c]quinoline Derivatives as Potent Dual Antituberculosis and Anti-Inflammatory Agents. *Molecules*, 22(6), 1001. <https://doi.org/10.3390/molecules22061001>
- Tugcu, G., Sipahi, H., & Aydin, A. (2019). Application of a Validated QSTR Model for Repurposing COX-2 Inhibitor Coumarin Derivatives as Potential Antitumor Agents. *Current Topics in Medicinal Chemistry*, 19(13), 1121–1128. <https://doi.org/10.2174/1568026619666190618143552>

- Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, *85*, 189–203. <https://doi.org/10.1016/j.jbi.2018.07.014>
- Urias, R. W. P., Barigye, S. J., Marrero-Ponce, Y., García-Jacas, C. R., Valdes-Martini, J. R., & Perez-Gimenez, F. (2015). IMMAN: free software for information theory-based chemometric analysis. *Molecular Diversity*, *19*(2), 305–319. <https://doi.org/10.1007/s11030-014-9565-z>
- Valdés-Martini, J. R., Marrero-Ponce, Y., García-Jacas, C. R., Martinez-Mayorga, K., Barigye, S. J., Vaz d'Almeida, Y. S., Pham-The, H., Pérez-Giménez, F., & Morell, C. A. (2017). QuBiLS-MAS, open source multi-platform software for atom- and bond-based topological (2D) and chiral (2.5D) algebraic molecular descriptors computations. *Journal of Cheminformatics*, *9*(1), 35. <https://doi.org/10.1186/s13321-017-0211-5>
- Vall, A., Sabnis, Y., Shi, J., Class, R., Hochreiter, S., & Klambauer, G. (2021). The Promise of AI for DILI Prediction. *Frontiers in Artificial Intelligence*, *4*, 15. <https://doi.org/10.3389/frai.2021.638410>
- Wang, D., Shi, J., Xiong, Y., Hu, J., Lin, Z., Qiu, Y., & Cheng, J. (2018). A QSAR-based mechanistic study on the combined toxicity of antibiotics and quorum sensing inhibitors against *Escherichia coli*. *Journal of Hazardous Materials*, *341*, 438–447. <https://doi.org/10.1016/j.jhazmat.2017.07.059>
- Wang, J., Yun, D., Yao, J., Fu, W., Huang, F., Chen, L., Wei, T., Yu, C., Xu, H., Zhou, X., Huang, Y., Wu, J., Qiu, P., & Li, W. (2018). Design, synthesis and QSAR study of novel isatin analogues inspired Michael acceptor as potential anticancer compounds. *European Journal of Medicinal Chemistry*, *144*, 493–503. <https://doi.org/10.1016/j.ejmech.2017.12.043>
- WEKA software. (2023, agosto 2). <https://www.cs.waikato.ac.nz/ml/weka/>
- Wiegand, R. P., & Jong, K. A. (2004). *An Analysis of Cooperative Coevolutionary Algorithms* [PhD]. George Mason University. <https://dl.acm.org/doi/10.5555/997339>
- Willighagen, E. L., Mayfield, J. W., Alvarsson, J., Berg, A., Carlsson, L., Jeliakova, N., Kuhn, S., Pluskal, T., Rojas-Chertó, M., Spjuth, O., Torrance, G., Evelo, C. T., Guha, R., & Steinbeck, C. (2017). The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of Cheminformatics*, *9*(1), 33. <https://doi.org/10.1186/s13321-017-0220-4>
- Wishart, D., Arndt, D., Pon, A., Sajed, T., Guo, A. C., Djoumbou, Y., Knox, C., Wilson, M., Liang, Y., Grant, J., Liu, Y., Goldansaz, S. A., & Rappaport, S. M. (2015). T3DB: the toxic exposome database. *Nucleic Acids Research*, *43*(D1), D928–D934. <https://doi.org/10.1093/nar/gku1004>
- Wishart, D. S. (2007). Introduction to Cheminformatics. En *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc. <https://doi.org/10.1002/0471250953.bi1401s18>
- Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., Assempour, N., Iynkkaran, I., Liu, Y., Maclejewski, A., Gale, N., Wilson, A., Chin, L., Cummings, R., Le, D., ... Wilson, M. (2018). DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Research*, *46*(D1), D1074–D1082. <https://doi.org/10.1093/nar/gkx1037>

- Xing, J.-J., Luo, R.-M., Guo, H.-L., Li, Y.-Q., Fu, H.-Y., Yang, T.-M., & Zhou, Y.-P. (2014). Radial basis function network-based transformation for nonlinear partial least-squares as optimized by particle swarm optimization: Application to QSAR studies. *Chemometrics and Intelligent Laboratory Systems*, 130, 37–44. <https://doi.org/10.1016/j.chemolab.2013.10.006>
- Xiong, G., Wu, Z., Yi, J., Fu, L., Yang, Z., Hsieh, C., Yin, M., Zeng, X., Wu, C., Lu, A., Chen, X., Hou, T., & Cao, D. (2021). ADMETlab 2.0: An integrated online platform for accurate and comprehensive predictions of ADMET properties. *Nucleic Acids Research*, 49(W1). <https://doi.org/10.1093/nar/gkab255>
- Xu, Y., Dai, Z., Chen, F., Gao, S., Pei, J., & Lai, L. (2015). Deep Learning for Drug-Induced Liver Injury. *Journal of Chemical Information and Modeling*, 55(10), 2085–2093. <https://doi.org/10.1021/acs.jcim.5b00238>
- Xu, Y., Pei, J., & Lai, L. (2017). Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *Journal of Chemical Information and Modeling*, 57(11), 2672–2685. <https://doi.org/10.1021/acs.jcim.7b00244>
- Yap, C. W. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, 32(7), 1466–1474. <https://doi.org/10.1002/jcc.21707>
- Yousefinejad, S., & Hemmateenejad, B. (2015). Chemometrics tools in QSAR/QSPR studies: A historical perspective. *Chemometrics and Intelligent Laboratory Systems*, 149, 177–204. <https://doi.org/10.1016/j.chemolab.2015.06.016>
- Marrero, Y. (2023a, julio 30). *SiLiS-PREENZA*. <http://tomocomd.com/>.
- Marrero, Y. (2023b, julio 30). *SiLiS-PTOXRA*. <http://tomocomd.com/>.
- Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1), 3–28. [https://doi.org/10.1016/0165-0114\(78\)90029-5](https://doi.org/10.1016/0165-0114(78)90029-5)

Anexos

Tabla 22. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de prueba Liew_1_R_TS_120 y Liew_2_B_TS_47, así como en sus respectivos DA

Modelo	Liew_1_R_TS_120				Liew_2_B_TS_47			
	ACC	Sen	Spe	MCC	ACC	SEN	SPE	MCC
L1	0.79	0.89	0.65	0.56	0.85	0.91	0.79	0.71
	0.78(5,0.96) ^a	0.88	0.64	0.54	0.85(1,0.98)	0.91	0.79	0.70
L2	0.78	0.89	0.63	0.54	0.85	0.91	0.79	0.71
	0.78(4,0.97)	0.90	0.61	0.54	0.85(1,0.98)	0.91	0.79	0.70
L3	0.78	0.90	0.60	0.54	0.85	0.91	0.79	0.71
	0.79(3,0.98)	0.91	0.60	0.56	0.85(1,0.98)	0.91	0.79	0.70
N1	0.97	0.99	0.94	0.93	1.00	1.00	1.00	1.00
N2	0.95	0.97	0.92	0.90	1.00	1.00	1.00	1.00
N3	0.96	0.99	0.92	0.91	1.00	1.00	1.00	1.00
S1	0.96	0.94	0.98	0.92	1.00	1.00	1.00	1.00
S2	0.96	0.94	0.98	0.92	1.00	1.00	1.00	1.00
S3	0.96	0.94	0.98	0.92	1.00	1.00	1.00	1.00
<u>M2 (Base)</u>	0.78	0.92	0.56	0.53	0.87	0.96	0.79	0.76
<u>M9 (Base)</u>	0.77	0.86	0.63	0.51	0.85	0.91	0.79	0.71
<i>E13 (Ensemble)</i>	0.79	0.89	0.65	0.56	0.94	0.96	0.92	0.87
<i>E12 (Ensemble)</i>	0.79	0.89	0.65	0.56	0.94	0.96	0.92	0.87
Liew Ensemble	0.75	0.82	0.65	0.47	0.81	0.96	0.67	0.65
Liew Base	0.71	0.68	0.75	0.42	0.83	0.83	0.83	0.66
DL-Liew (Dili server)	0.67	0.60	0.71	0.31	0.62	0.78	0.46	0.25
DL-Combined (Dili server)	0.58	0.69	0.51	0.20	0.77	0.83	0.70	0.54
Vslead	0.83	0.93	0.67	0.63	0.72	0.87	0.58	0.47
Padel predictor	0.76	0.65	0.83	0.49	0.81	0.67	0.96	0.65
pkCSM	0.51	0.60	0.44	0.05	0.53	0.71	0.35	0.06
AdmetLab	0.66	0.69	0.64	0.32	0.79	0.79	0.78	0.57
AdmetLab 2.0	0.64	0.56	0.69	0.26	0.79	0.83	0.74	0.58
vNN-Admet	0.88	0.85	0.92	0.76	0.96	0.96	0.96	0.91
DNN-ECFP4	0.60	0.65	0.52	----	0.83	0.96	0.71	----
Mold2+DeepDILI	0.66	0.76	0.50	0.27	0.94	0.96	0.92	0.87
DILI-CNN-MFE-I	0.78	0.60	0.89	0.52	0.74	0.50	1.00	0.57
DILI-CNN-MFE-II	0.87	0.85	0.88	0.72	0.91	0.83	1.00	0.84
DILI-CNN-MFE-III	0.86	0.85	0.86	0.71	0.98	0.96	1.00	0.96
DILI-CNN-MFE-IV	0.65	0.27	0.90	0.23	0.68	0.38	1.00	0.48

^a: La segunda fila por modelo es el rendimiento del modelo considerando los compuestos dentro del dominio de aplicabilidad. Entre paréntesis se muestra el número de compuestos que no caen dentro, así como el % de cobertura. Para los modelos N1, N2, N3, S1, S2, S3 todos los compuestos caen dentro del DA.

Tabla 23. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de prueba Mora_4_ETS_554 y Nguyen_1_TS_322, así como en sus respectivos DA

Model Id	Mora_4_ETS_554				Nguyen_1_TS_322			
	ACC	SEN	SPE	MCC	ACC	SEN	SPE	MCC
L1	0.78	0.83	0.67	0.50	0.86	0.95	0.80	0.74
	0.78(12,0.98) ^a	0.84	0.68	0.51	0.86(4,0.99)	0.95	0.80	0.74
L2	0.78	0.84	0.67	0.50	0.86	0.96	0.80	0.74
	0.78(11,0.98)	0.84	0.66	0.50	0.86(4,0.99)	0.96	0.80	0.75
L3	0.78	0.84	0.66	0.50	0.86	0.98	0.79	0.75
	0.78(10,0.98)	0.84	0.67	0.10	0.86(4,0.99)	0.98	0.79	0.75
N1	0.88	0.87	0.91	0.75	0.92	0.89	0.98	0.85
	0.88(3,0.99)	0.87	0.91	0.75				
N2	0.89	0.88	0.90	0.76	0.92	0.89	0.98	0.85
	0.89(3,0.99)	0.88	0.91	0.76				
N3	0.88	0.87	0.90	0.75	0.92	0.89	0.98	0.85
	0.88(3,0.99)	0.87	0.91	0.75				
S1	0.82	0.79	0.89	0.64	0.93	0.98	0.90	0.87
					0.95(9,0.97)	0.98	0.93	0.90
S2	0.83	0.79	0.91	0.66	0.93	0.98	0.91	0.88
					0.95(9,0.97)	0.98	0.93	0.90
S3	0.82	0.78	0.91	0.64	0.94	0.98	0.91	0.88
	0.82(1,0.99)	0.78	0.91	0.64	0.95(10,0.97)	0.98	0.93	0.91
<u>M2 (Base)</u>	0.77	0.84	0.61	0.46	0.80	0.96	0.68	0.64
<u>M9 (Base)</u>	0.76	0.80	0.66	0.45	0.72	0.95	0.57	0.54
<i>E13 (Ensemble)</i>	0.78	0.82	0.70	0.51	0.84	0.95	0.76	0.70
<i>E12 (Ensemble)</i>	0.79	0.83	0.71	0.53	0.84	0.96	0.76	0.71
DL-Liew (Dili server)	0.67	0.65	0.68	0.30	----	----	----	----
DL-Combined (Dili server)	0.65	0.74	0.60	0.32	----	----	----	----
Vslead	0.79	0.88	0.60	0.50	----	----	----	----
Padel predictor	0.74	0.62	0.79	0.41	0.85	0.82	0.89	0.69
pkCSM	0.58	0.57	0.59	0.15	0.56	0.61	0.53	0.15
AdmetLab	0.70	0.78	0.67	0.42	0.81	0.86	0.77	0.62
AdmetLab 2.0	0.70	0.71	0.69	0.38	0.82	0.88	0.79	0.65
vNN-Admet	0.78	0.92	0.72	0.60	0.94	0.99	0.91	0.89
Mold2+DeepDILI	0.73	0.74	0.71	0.43	0.87	0.93	0.83	0.74
DILI-CNN-MFE-I	0.82	0.64	0.91	0.57	0.77	0.98	0.63	0.62
DILI-CNN-MFE-II	0.86	0.85	0.86	0.69	0.86	0.96	0.80	0.75
DILI-CNN-MFE-III	0.85	0.88	0.84	0.69	0.88	0.95	0.84	0.77
DILI-CNN-MFE-IV	0.77	0.44	0.93	0.44	0.63	0.98	0.38	0.43

^a: La segunda fila por modelo es el rendimiento del modelo considerando los compuestos dentro del dominio de aplicabilidad. Entre paréntesis se muestra el número de compuestos que no caen dentro, así como el % de cobertura. Para los modelos N1, N2, N3 los todos compuestos del conjunto Nguyen_1_TS_322 caen dentro del DA, mientras que para los modelos S1, S2 todos los compuestos del conjunto MORA_4_TS_554 caen dentro del DA.

Tabla 24. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en los conjuntos de Nguyen_2_TS_52 y Garcia_1_TS_106, así como en sus respectivos DA

Modelo	Nguyen_2_TS_52				Garcia_1_TS_106			
	ACC	SEN	SPE	MCC	ACC	SEN	SPE	MCC
L1	0.85	1.00	0.70	0.73	0.79	0.93	0.55	0.53
	0.84(1,0.98) ^a	1.00	0.69	0.72	0.80(1,0.99)	0.94	0.55	0.56
L2	0.87	1.00	0.74	0.76	0.79	0.93	0.55	0.53
	0.86(1,0.98)	1.00	0.73	0.76	0.80(1,0.99)	0.94	0.55	0.56
L3	0.85	1.00	0.70	0.73	0.81	0.93	0.61	0.58
	0.84(1,0.98)	1.00	0.69	0.72	0.82(1,0.99)	0.94	0.61	0.60
N1	0.92	1.00	0.85	0.86	0.94	0.90	0.97	0.88
N2	0.92	1.00	0.85	0.86	0.96	0.92	0.99	0.92
N3	0.92	1.00	0.85	0.86	0.96	0.92	0.99	0.92
S1	0.94	0.96	0.93	0.89	0.92	0.91	0.92	0.82
	0.94(2,0.96)	0.96	0.92	0.88				
S2	0.94	0.96	0.93	0.89	0.92	0.91	0.95	0.84
	0.94(2,0.96)	0.96	0.92	0.88				
S3	0.94	0.96	0.93	0.89	0.92	0.91	0.95	0.94
	0.94(2,0.96)	0.96	0.92	0.88	0.91(1,0.99)	0.91	0.92	0.92
<u>M2 (Base)</u>	0.84	0.78	0.92	0.70	0.81	0.96	0.55	0.58
<u>M9 (Base)</u>	0.73	0.48	1.00	0.56	0.85	0.93	0.74	0.69
<i>E13 (Ensemble)</i>	0.84	0.78	0.92	0.70	0.82	0.91	0.66	0.60
<i>E12 (Ensemble)</i>	0.83	0.70	0.96	0.68	0.82	0.91	0.66	0.60
Padel predictor	0.81	0.84	0.77	0.62	0.77	0.58	0.88	0.49
pkCSM	0.60	0.76	0.44	0.21	0.60	0.62	0.58	0.19
AdmetLab	0.75	0.84	0.67	0.51	0.77	0.78	0.76	0.53
AdmetLab 2.0	0.83	0.88	0.78	0.66	0.79	0.63	0.88	0.54
vNN-Admet	0.94	0.89	1.00	0.89	0.85	0.95	0.79	0.71
Mold2+DeepDILI	0.75	0.80	0.70	0.51	0.83	0.90	0.71	0.62
DILI-CNN-MFE-I	0.77	1.00	0.56	0.61	0.90	0.74	0.99	0.78
DILI-CNN-MFE-II	0.81	1.00	0.63	0.67	0.94	0.87	0.99	0.88
DILI-CNN-MFE-III	0.83	0.96	0.70	0.68	0.94	0.87	0.99	0.88
DILI-CNN-MFE-IV	0.60	1.00	0.22	0.35	0.89	0.71	0.99	0.76

^a: La segunda fila por modelo es el rendimiento del modelo considerando los compuestos dentro del dominio de aplicabilidad. Entre paréntesis se muestra el número de compuestos que no caen dentro, así como el % de cobertura. Para los modelos N1, N2, N3 los todos compuestos del conjunto Nguyen_1_TS_322 caen dentro del DA, mientras que para los modelos S1, S2 todos los compuestos del conjunto MORA_4_TS_554 caen dentro del DA.

Tabla 25. Comparación del rendimiento de los nueve modelos cooperativos respecto a diferentes modelos y herramientas disponibles en la literatura para la predicción de DILI en el conjunto Liew_3_ValPair_20, así como en su respectivo DA.

Model Id	Liew_3_ValPair_20			
	ACC	SEN	SPE	MCC
L1	0.55	0.80	0.30	0.12
L2	0.55	0.90	0.20	0.14
L3	0.55	0.70	0.40	0.11
N1	0.65	0.70	0.60	0.30
N2	0.70	0.70	0.70	0.40
N3	0.65	0.70	0.60	0.30
S1	0.55	0.70	0.40	0.11
S2	0.55	0.70	0.40	0.11
S3	0.55	0.70	0.40	0.11
<u>M2 (Base)</u>	0.55	0.30	0.80	0.12
<u>M9 (Base)</u>	0.50	0.30	0.70	0.00
<i>E13 (Ensemble)</i>	0.50	0.30	0.70	0.00
<i>E12 (Ensemble)</i>	0.50	0.30	0.70	0.00
Liew Ensemble	0.55	0.80	0.30	0.12
Liew Base	0.50	0.70	0.30	0.00
DL-Liew (Dili server)	0.45	0.10	0.80	-0.14
DL-Combined (Dili server)	0.55	0.50	0.50	0.10
Padel predictor	0.55	0.30	0.80	0.12
pkCSM	0.45	0.70	0.20	-0.12
AdmetLab	0.55	0.60	0.50	0.10
AdmetLab 2.0	0.50	0.50	0.50	0.00
vNN-Admet	0.55	0.50	0.60	0.10
DNN-ECFP4	0.45	0.70	0.20	----
Mold2+DeepDILI	0.50	0.50	0.50	0.00
DILI-CNN-MFE-I	0.55	0.50	0.60	0.10
DILI-CNN-MFE-II	0.60	0.70	0.50	0.20
DILI-CNN-MFE-III	0.65	0.90	0.40	0.35
DILI-CNN-MFE-IV	0.55	0.50	0.60	0.10

Para los nueve cooperativos, todos los compuestos caen dentro del DA.

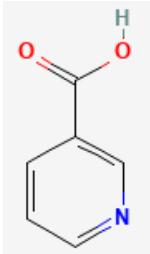
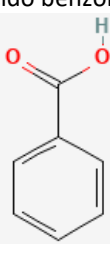
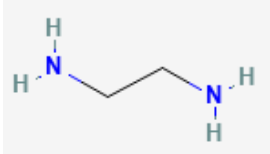

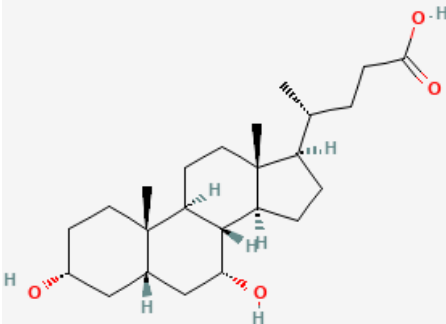
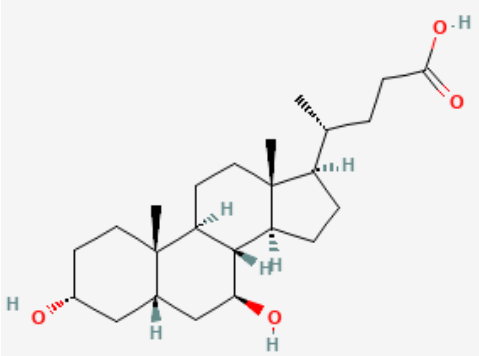
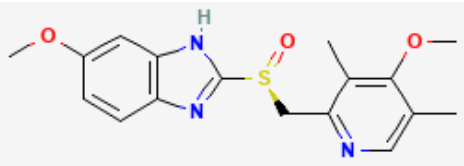
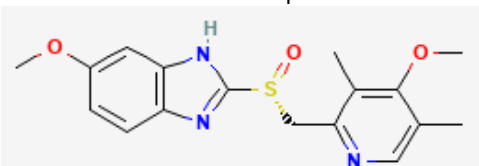
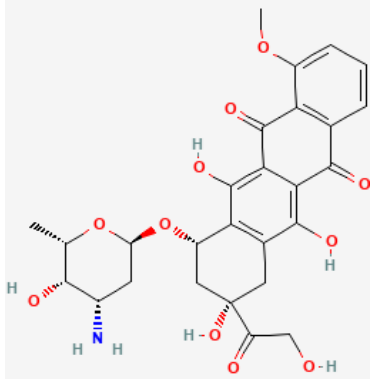
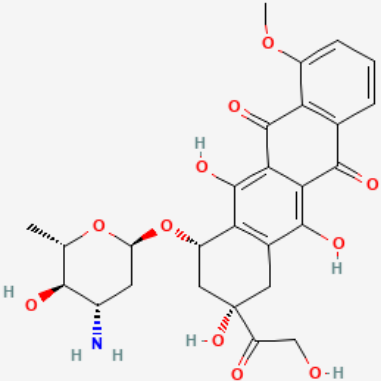
Par	Compuesto hepatóxico	Compuesto no hepatóxico
1	<p>Niacina</p>  <p>PID: 938</p>	<p>Ácido benzoico</p>  <p>PID: 243</p>
2	<p>etilendiamina</p>  <p>PID: 3301</p>	<p>etanolamina</p>  <p>PID: 700</p>
3	<p>ácido quenodesoxicólico</p>  <p>PID: 10133</p>	<p>ácido ursodesoxicólico</p>  <p>PID:31401</p>
4	<p>S-Omeprazole</p>  <p>PID: 9568614</p>	<p>R-Omeprazole</p>  <p>PID: 9579578</p>
5	<p>Doxorubicina</p>  <p>PID: 31703</p>	<p>epirubicina</p>  <p>PID: 41867</p>

Figura 13. Estructura química de los compuestos incluidos en el conjunto de datos Liew_3_ValPair_20

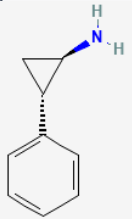
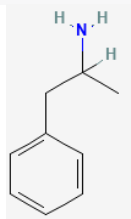
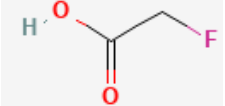
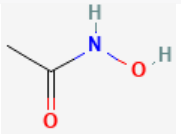
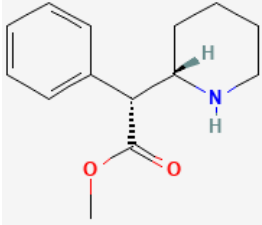
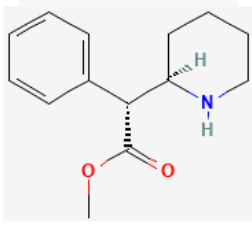
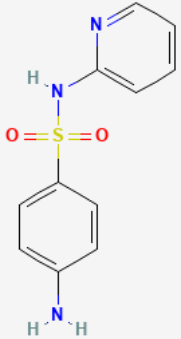
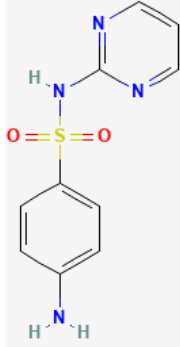
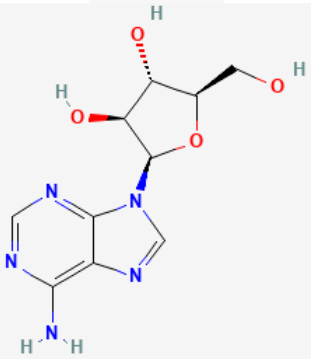
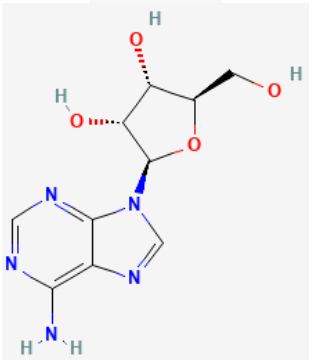
Par	Compuesto hepatóxico	Compuesto no hepatóxico
6	<p>Tranilcipromina (Parnate)</p>  <p>PID: 19493</p>	<p>anfetamina</p>  <p>PID: 3007</p>
7	<p>ácido fluoroacético</p>  <p>PID: 5237</p>	<p>ácido acetohidroxámico</p>  <p>PID: 1990</p>
8	<p>metilfenidato</p>  <p>PID: 3040629</p>	<p>dexmetilfenidato</p>  <p>PID:154101</p>
9	<p>sulfapiridina</p>  <p>PID: 5336</p>	<p>sulfadiazina</p>  <p>PID: 5215</p>
10	<p>vidarabina</p>  <p>PID: 21704</p>	<p>adenosina</p>  <p>PID: 60961</p>

Figura 14. Estructura química de los compuestos incluidos en el conjunto de datos Liew_3_ValPair_20

Código fuente 1. Ejemplo de clase para la definición del cromosoma que representa al DM RDF

```
public class RDFHead implements AMDHead {
    private String radio;
    private String prop;
    private String b;
    public RDFHead() {
        setName("RDF");
        randomHeading();
    }
    @Override
    public MDType getType() {
        return MDType.RDF;
    }
    @Override
    public String randomHeading() {
        prop = GetRdfPropertiesParam();
        b = GetRDFSsmoothingParam();
        radio = GetRadioParam();
        return toString();
    }
}
```

Código fuente 2. Código fuente para el cálculo del DM RDF

```
public class MDRdf implements IMDComputer {
    @Override
    public TomocomdInstances compute(Set<String> descSet,
        String sdfPath) {
        TomocomdInstances data = null;
        try {
            for (String desc : descSet) {
                TomocomdInstances descData
                    = computeDesc4Sdf(desc, sdfPath);
                if (data == null) {
                    data = new TomocomdInstances(descData);
                } else {
                    data = TomocomdInstances.merge(data, descData);
                }
            }
            return data;
        } catch (TomocomdIOException ex) {
            throw new MDComputerException("Problems loading sdf",ex);
        } catch (Exception ex) {
            throw new MDComputerException(
                "Problems computing RDF MDs",ex);
        }
    }
}
```


Código fuente 3. Código fuente ejemplo para la ejecución del algoritmo cooperativo

```
1. GAConf conf = new GAConf();
2. conf.setNumIter(2000);
3. conf.setCoop(true);
4. conf.getGAResetConf().setNumIter(20);
5.
6. conf.getGaAlgorithm4PobConf().setNumDesc(50);
7. conf.getGaAlgorithm4PobConf().getSelConf().setCant(4);
8. conf.getGaAlgorithm4PobConf().getFiltersConfig().add(
    new FilterConfig(FilterType.SE, new String[]{"-
    t", "0.1"}));

9. String OUT_FILE_NAME = "best_subset.csv";
10. String SDF_MOL_FILE = "ace_COR3D_training.sdf";
11. String ENDPOINT = "activity";

12. conf.getAmdHeadList().clear();
13. conf.getAmdHeadList().add(new QuadraticMasHead());
14. conf.getAmdHeadList().add(new BilinealMasHead());
15. conf.getAmdHeadList().add(new LinearMasHead());
16.
17. try {
    AGeneticAlgorithm algorithm = new MultiCoreGA(conf,
        OUT_FILE_NAME, ENDPOINT, SDF_MOL_FILE);
18.     algorithm.compute();
19. } catch (GAExecutionException e) {
20.     System.exit(-1);
21. }
```

Publicaciones relacionadas

1. **L. A. García-González**, Y. Marrero-Ponce, C. A. Brizuela, and C. R. García-Jacas, "Overproduce and select, or determine optimal molecular descriptor subset via configuration space optimization? Application to the prediction of ecotoxicological endpoints," *Mol Inform*, vol. 42, no. 6, 2023, doi: 10.1002/minf.202200227
2. **L. A. García-González**, Y. Marrero-Ponce, C. A. Brizuela, and C. R. García-Jacas, "A Chemical Dataset- and Endpoint-guided Co-Evolutionary Approach to Determine Optimal Molecular Descriptor Sets. Application to the Drug-Induced Liver Injury (DILI) prediction". *Mol Diversity* (En revisión)

Colaboraciones

1. Carballo GM, Vázquez KG, **García-González LA**, Rio GD, Brizuela CA. Embedded-AMP: A Multi-Thread Computational Method for the Systematic Identification of Antimicrobial Peptides Embedded in Proteome Sequences. *Antibiotics*. 2023; 12(1):139
2. César R García-Jacas, **Luis A García-González**, Felix Martinez-Rios, Issac P Tapia-Contreras, Carlos A Brizuela, Handcrafted versus non-handcrafted (self-supervised) features for the classification of antimicrobial peptides: complementary or redundant?, *Briefings in Bioinformatics*, Volume 23, Issue 6, November 2022, bbac428
3. César R García-Jacas, Sergio A Pinacho-Castellanos, **Luis A García-González**, Carlos A Brizuela, Do deep learning models make a difference in the identification of antimicrobial peptides?, *Briefings in Bioinformatics*, Volume 23, Issue 3, May 2022, bbac094
4. Fonseca MC, Pupo-Meriño M, **García-González LA**, Muné M, Resik S, Norder H, Sarmiento L. Molecular Characterization of Coxsackievirus A24v from Feces and Conjunctiva Reveals Epidemiological Links. *Microorganisms*. 2021; 9(3):531.
5. Fonseca, M.C., Pupo-Meriño, M., **García-González, L.A. et al.** Molecular evolution of coxsackievirus A24v in Cuba over 23-years, 1986–2009. *Sci Rep* **10**, 13761 (2020).

Premios



El Pleno de la Academia de Ciencias de Cuba

en uso de las atribuciones que le confiere el Decreto-Ley 163 de 1996 y, con el propósito de reconocer los resultados de las investigaciones que se destaquen en el país, adoptó el siguiente

ACUERDO

PRIMERO: Conceder uno de sus Premios Anuales del año 2021 al resultado de la investigación científica denominado:

Caracterización molecular del CVA24v aislado en Cuba durante cinco períodos epidémicos de conjuntivitis hemorrágica aguda, revela hallazgos en la epidemiología y patogenia del virus

De la entidad ejecutora principal: Instituto de Medicina Tropical Dr. Pedro Kourí.

Con la participación de las entidades: Departamento de Bioinformática, Universidad de las Ciencias Informáticas; Centro de Estudios de Matemática Computacional, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas; Unidad de Immunovirología, Departamento de Ciencias Clínicas, Hospital de la Universidad Skåne, Universidad de Lund- Suecia.

SEGUNDO: A todos los efectos de autoría del resultado premiado y acorde a la propuesta recibida, reconocer a las personas que se relacionan:

Autora principal: Magilé C. Fonseca Quintana.

Otros autores: Mario Pupo Meriño, Luis Sarmiento Pérez, Luis A. García González, Sonia Resik Aguirre, Lai Heng Hung, Mayra Muné.

TERCERO: Otorgar en acto público el correspondiente diploma que certifica lo anterior a las autoridades de las entidades donde fue obtenido el resultado premiado, a través de sus autores, a quienes entregamos este acuerdo.

Y para que así conste, se emite el presente, con fecha de mayo de 2022, Año 62 de la Revolución.

Dr.Cs. Luis C. Velázquez Pérez
Presidente



43 / 2021 Ciencias Biomédicas

Reg. #: 104

Exptd. #: 2987