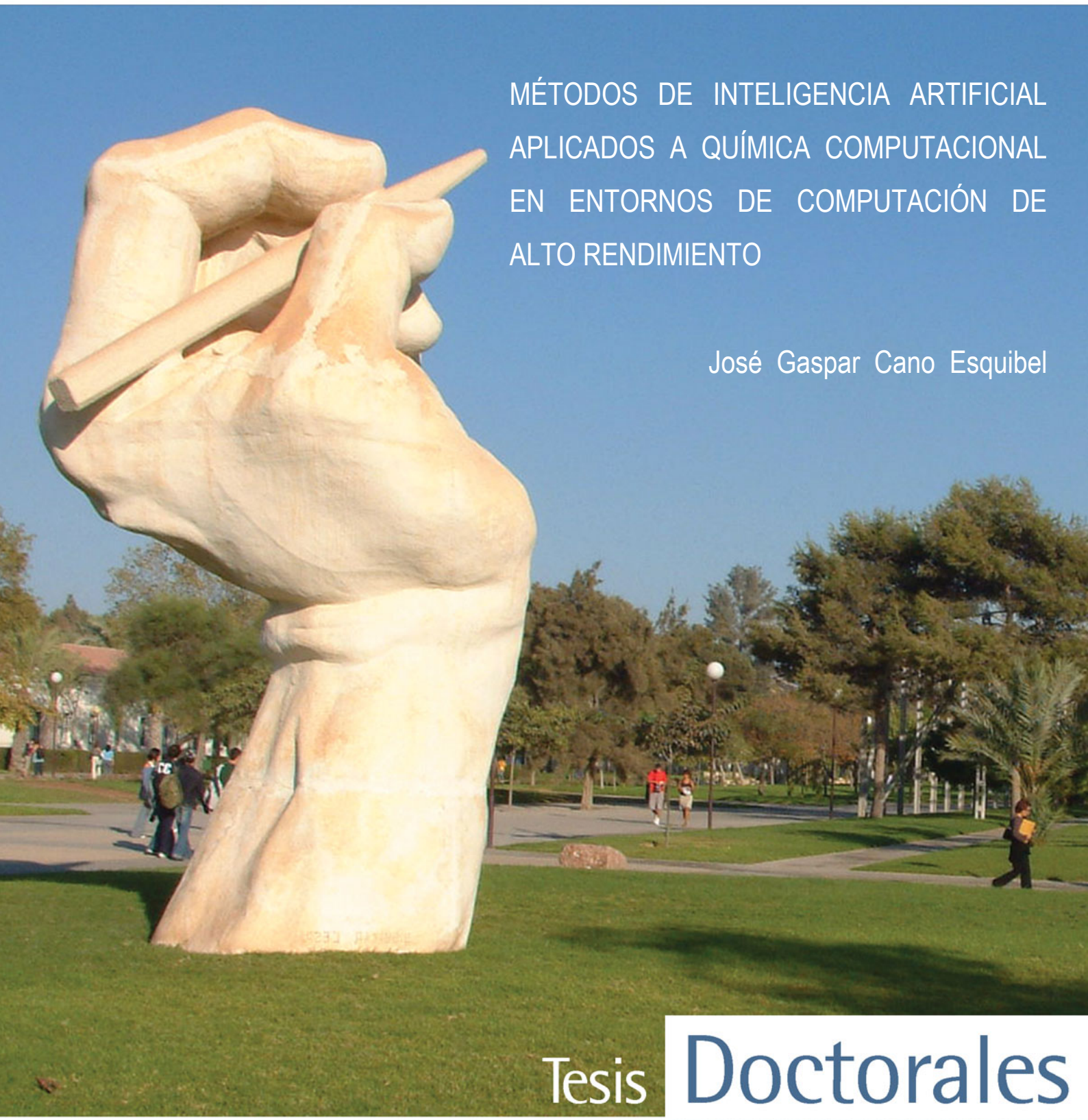
Universitat d'Alacant
Universidad de Alicante

MÉTODOS DE INTELIGENCIA ARTIFICIAL APLICADOS A QUÍMICA COMPUTACIONAL EN ENTORNOS DE COMPUTACIÓN DE ALTO RENDIMIENTO

José Gaspar Cano Esquibel



Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

UNIVERSIDAD DE ALICANTE

Tesis Doctoral

MÉTODOS DE INTELIGENCIA ARTIFICIAL APLICADOS A QUÍMICA COMPUTACIONAL EN ENTORNOS DE COMPUTACIÓN DE ALTO RENDIMIENTO

José Gaspar Cano Esquibel

Dirigida por:

Dr. José García Rodríguez

Dr. Horacio Pérez Sánchez

Diciembre 2014

TENOLOGÍAS PARA LA SOCIEDAD DE LA INFORMACIÓN

No tenemos la oportunidad de hacer muchas cosas, por lo que cada cosa que hagamos debe ser excelente. Porque esta es nuestra vida.

Steve Jobs.

AGRADECIMIENTOS

Resulta difícil recordar los nombres de todas las personas que han influido en mi y sería injusto olvidarme de cualquiera de ellos.

En primer lugar, me gustaría agradecer enormemente a mis directores de tesis José y Horacio, por su inagotable dedicación durante el desarrollo de este trabajo, sin ellos nada de esto hubiera sido posible. Gracias por compartir conmigo todos vuestros conocimientos y experiencias, por haberme enseñado a mejorar cada día en mi trabajo. También agradeceros vuestra relación más allá de lo profesional y por haberme tratado como un amigo, y a mi edad ya no se hacen amigos.

Gracias a mi Familia y en especial a mis hijos, Jaime y Paula, por el tiempo que les he robado y sin casi cuestionar porque lo hacia.

A mi amigo Gabriel López, que empezamos juntos la carrera hace ya tanto tiempo y los dos presentamos la tesis este mismo año.

Agradezco sinceramente a mis compañeros y amigos del Departamento de Tecnología Informática y Computación de la Universidad de Alicante, y de la Escuela Politécnica Superior de Alicante, así como al grupo "*Computer Science Department*" de la Universidad Católica de Murcia (UCAM) por estar ahí.

Y por supuesto a mis amigos de siempre..

José Gaspar Cano Esquibel

Alicante, 12 de Noviembre de 2.014

RESUMEN

Uno de los problemas científicos más importantes actualmente, y que concentra mayores esfuerzos investigadores en los últimos años, es el descubrimiento de nuevos compuestos bioactivos para resolver problemas de relevancia biológica o donde los compuestos conocidos previamente no son lo suficientemente efectivos. Tradicionalmente ha sido la industria farmacéutica quien se ha ocupado del estudio de dichos problemas, debido al gran coste económico que implica y a sus dificultades técnicas. Desde hace unas dos décadas existen metodologías basadas en la aplicación de técnicas de modelado molecular que permiten acelerar dichos descubrimientos, y que pueden ser desarrolladas de manera eficiente en un entorno académico a un coste mucho menor. Como consecuencia, es posible acelerar drásticamente mediante simulación por ordenador dichos procesos de descubrimiento de compuestos bioactivos, cuando se mezcla una investigación multidisciplinar (Química, Biología, Informática, Ingeniería, Medicina) con la explotación de supercomputadores y arquitecturas paralelas de

alto rendimiento. No obstante, tanto esta metodología como otras usadas dentro del mismo campo por decenas de miles de investigadores, todavía presentan una serie de limitaciones a nivel de predicción y de velocidad de proceso de datos, dos factores de extrema relevancia para poder llevar a cabo con éxito la investigación biomédica.

El principal objetivo de esta tesis es por tanto: la propuesta de una serie de refinamientos basados en métodos de inteligencia computacional, unidos a metodología in-silico para el descubrimiento de compuestos bioactivos, gracias a la capacidad de cómputo proporcionada por la reciente aparición de las arquitecturas computacionales masivamente paralelas tales como las GPUs (*Graphics Processing Units)*. Los resultados obtenidos en este proyecto podrían por tanto ayudar y formar la base de una nueva y atractiva generación de aproximaciones para descubrimiento de compuestos bioactivos.

ABSTRACT

One of the most important scientific problems currently, that concentrated researcher efforts in recent years, is the discovery of new bioactive compounds to solve problems of biological relevance or where the previously known compounds are not effective enough. It has traditionally been the pharmaceutical industry who has worked on the study of such problems, due to the large economic cost involved and their technical difficulties. Since about two decades ago there are methodologies based on the application of molecular modeling techniques that can help accelerate these discoveries, and that can be developed efficiently in an academic environment at a lower cost. As a result, it is possible to accelerate dramatically by computer simulation such processes of discovery of bioactive compounds, when you mix a multidisciplinary research (chemistry, biology, computer science, engineering, and medicine) with the exploitation of supercomputers and parallel architectures for high performance. However, both, this methodology as other used within the same field by tens of thousands of researchers, have still a number of limitations on the level of predictive capacity and speed of processing data, two factors of extreme relevance in order to carry out successful projects of this type.

The main objective of this thesis is therefore: the proposal of a series of refinements based on methods of computational intelligence in cooperation with in-silico methodology for the discovery of bioactive compounds, thanks to the computing capacity provided by the recent emergence of the massively parallel computational architectures such as the GPUs (*Graphics Processing Units)*. The results obtained in this project could therefore help and form the basis of a new and attractive generation of approximations for discovery of bioactive compounds.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

El trabajo que presenta este documento es fruto de la investigación desarrollada a lo largo de los últimos años en el campo de la inteligencia computacional en mi colaboración con el departamento de Tecnología Informática y Computación de la Universidad de Alicante, y en química computacional, con el grupo de Bioinformática y Computación de Altas Prestaciones, de la Universidad Católica de Murcia (UCAM).

En este primer capítulo hablaremos de la motivación y los objetivos que se pretenden alcanzar. Como paso previo a la exposición de la propuesta de solución al finalizar el capitulo y con el fin de analizar la problemática y estudiar las soluciones disponibles, se realiza una revisión del estado

del arte en los campos directamente relacionados. Por una parte, se han revisado trabajos sobre la química computacional, relacionados con el descubrimiento de fármacos. Por otra parte se han repasado las técnicas de inteligencia computacional, se han expuestos diferentes metodologías para llevar a cabo la predicción o clasificación requerida, así como la extracción mínima de características.

En lo referente a las restricciones temporales, se han estudiado arquitecturas para la predicción utilizando tecnologías masivamente paralelas de bajo coste.

Al final de este capítulo presentaremos la propuesta de solución a los requerimientos del problema a resolver y que satisfaga los objetivos que se pretenden alcanzar.

1.1 MOTIVACIÓN Y OBJETIVOS

En la investigación clínica, es crucial determinar la seguridad y eficacia de los fármacos actuales; así como acelerar significativamente la búsqueda de nuevos compuestos activos y la investigación clínica básica. Es decir, un gran número de hipótesis pueden ser analizadas antes de los ensayos clínicos, permitiendo abaratar costes al reducir el tiempo empleado en este importante proceso. Para cumplir estos objetivos se necesita procesar grandes bases de datos de estructuras de proteínas disponibles en bases de datos biológicos tales como PDB (*Protein Data Bank*) [1] y también de bases de datos genómicas [2] utilizando técnicas como el modelado de proteínas por homología.

Esto objetivo puede alcanzarse, gracias a la disponibilidad de las herramientas bioinformáticas y métodos de Cribado Virtual (CV, técnica computacional que se utiliza en el descubrimiento de fármacos para identificar los ligandos o moléculas pequeñas que tienen más probabilidades de unirse a una proteína diana). Los métodos de CV nos permiten probar todas las hipótesis requeridas antes de los ensayos clínicos. No obstante, los actuales métodos CV, tales como en el acoplamiento molecular (*Docking,* método que predice la colocación de un molécula, al unirse a otra, para formar un complejo estable). Otras referencias importantes de acoplamiento molecular son: GLIDE [3], AUTODOCK [4], DOCK [5] o BINDSURF [6]. Los métodos de CV no son absolutamente fiables en su predicción de toxicidad y actividad debido a sus limitaciones en el acceso a recursos de computación, por su elevada complejidad, incluso los métodos actuales más rápidos para el CV, no pueden procesar grandes bases de datos biológicos en un plazo de tiempo

razonable. Por lo tanto, estas restricciones imponen serias limitaciones en muchas áreas de la investigación.

Esta limitación se subsana es parte, gracias al uso de la última generación de arquitecturas de hardware masivamente paralelo, tales como las unidades de procesamiento gráfico (*GPU*). Estos dispositivos se han hecho cada vez más populares en el campo de la computación de alto rendimiento, al combinar una impresionante potencia de cálculo y los requisitos de los gráficos de tiempo real unido al lucrativo mercado de masas de la industria del videojuego [7]. De hecho, se han situado a la vanguardia de las arquitecturas masivamente paralelas. Estas arquitecturas están proporcionando excelentes aceleraciones en diversos tipos de aplicaciones, en comparación con las versiones secuenciales de las mismas, ejecutadas en plataformas que cuentan solo con unidades centrales de proceso (*CPU, Central Processing Units*), que han superado a estas en varias magnitudes en algunos casos [8]. Esto brinda una oportunidad única para incrementar la capacidad computacional de los ordenadores tradicionales, permitiendo así tener pequeños supercomputadores a precios muy asequibles.

Los científicos se han aprovechado de este dominio computacional y la GPU se ha convertido en un recurso clave en aplicaciones en las que el paralelismo es el denominador común [9]. Para mantener este impulso, las nuevas características de hardware, han sido añadidas progresivamente por NVIDIA [10] a su gama de GPUs, con la arquitectura Kepler [11], como hito más reciente. Por lo tanto, las GPU son muy adecuadas para superar esta falta de recursos computacionales en los métodos de CV, permitiendo la aceleración de los cálculos necesarios y la introducción de mejoras en los modelos biofísicos no asumibles en el pasado [12].

Se ha demostrado ampliamente que los métodos de CV pueden beneficiarse del uso de GPUs [6], [13], [14]. Por otra parte, una carencia importante en los métodos CV tradicionales es que, generalmente, asumen el lugar de unión ligando-proteína derivado de una sola estructura cristalina como común para los diferentes ligandos, algo que se ha demostrado que no siempre sucede [15], y por lo tanto es crucial evitar esta suposición básica.

Los últimos desarrollos en el campo de la computación de altas prestaciones vienen marcados por una clara consolidación del paralelismo como alternativa para afrontar los nuevos retos computacionales, principalmente impuesta por las limitaciones físicas del silicio [16]. Además, la evolución al paralelismo llega a su exponente máximo con las arquitecturas masivamente paralelas, que incluyen miles de cores simples dentro del chip [17] con el fin de incrementar el rendimiento de las aplicaciones; es por ello que estas plataformas también son conocidas como arquitecturas orientadas a *throughput* (número de aplicaciones procesadas por unidad de tiempo) [7]. Sin embargo, los métodos actuales de CV están limitados por la capacidad computacional necesaria para analizar grandes cantidades de datos de manera precisa y rápida, que los hagan aun mas atractivos al mundo científico [18]. El proceso requiere de tiempos de respuestas inmediatos, que permitan tomar decisiones en base a las hipótesis planteadas.

Por tanto, del análisis de las ultimas tendencias en computación de altas prestaciones (*HPC, High Performance Computing*), y la evaluación de sus principales ventajas y desventajas, se puede concluir que estos nuevos recursos computacionales van a pemitir el desarrollo de métodos de CV eficientes desde el punto de vista computacional, energético y económico y forman por ello parte de nuestra propuesta.

Así mismo, el uso de métodos de inteligencia computacional permitirá refinar los resultados de los métodos de CV y su inspiración biológica y paralelismo intrínseco, los hace candidatos ideales para aprovecharse, de igual modo de los recursos computacionales de altas prestaciones.

1.2 ESTADO DEL ARTE

En este apartado se presenta una revisión de los trabajos que abordan problemas similares al propuesto y que, teniendo relación con el problema planteado, puedan servir como referencia del estado en que se encuentra la investigación relacionada.

Se han revisado, por su relación con el marco en el que se encuentran, el desarrollo y la investigación de trabajos sobre inteligencia computacional, y de igual modo se revisan trabajos relacionados en el campo de la química computacional y la predicción de activad como refinamiento de las técnicas de cribado virtual.

1.1.1 QUÍMICA COMPUTACIONAL. DESCUBRIMIENTO DE FÁRMACOS

La integración de los últimos avances de investigación en los campos de la biología, química, física, matemáticas, medicina e informática, están permitiendo importantes avances en las áreas de la atención sanitaria, el descubrimiento de nuevos fármacos y la investigación genómica, entre otras. Estos avances están brindando nuevas estrategias terapéuticas, ofreciendo estilos de vida mas saludables que no eran imaginables hace tan solo unos años atrás. La unión de estos esfuerzos ha dado como fruto un nuevo campo de investigación multidisciplinar denominado Bioinformática [19] , que en líneas generales se puede definir como el uso de los últimos avances en informática para resolver los retos planteados en los campos de la biología y la medicina.

La solución a cualquier problema de química computacional necesita procesar grandes bases de datos de estructuras de proteínas disponibles en bases de datos biológicos tales como PDB [1] y también de bases de

datos genómicas [2] utilizando técnicas como el modelado de proteínas por homología [20] . La búsqueda de nuevos ligandos permite mejorar y encontrar soluciones a los tratamientos utilizados para diversas enfermedades y encontrar nuevas soluciones en el caso de enfermedades actualmente sin solución.

El método de cribado virtual de alto rendimiento por ordenador (*CAR*) [21], [22] permite identificar compuestos candidatos que se unan a una proteína diana con gran afinidad de entre millones de compuestos químicos disponibles en bases de datos públicas o privadas. Esto se consigue mediante la determinación de la posición óptima de acoplamiento del ligando con respecto a la proteína y el cálculo de la intensidad de interacción con la proteína. Aquellos compuestos con las mayores afinidades de unión se seleccionan para las siguientes fases de refinamiento estructural molecular y posibles estudios in-vitro, en animales y finalmente en ensayos clínicos en humanos. Aunque los métodos de CV han sido investigados desde hace unas dos décadas y se han descubierto varios compuestos que finalmente se han convertido en fármacos, éstos no son todavía lo suficientemente precisos para identificar de manera general y sistemática ligandos con alta afinidad por las proteínas. Para poder procesar grandes librerías con millones de compuestos, los métodos CV deben ser lo suficientemente rápidos para poder realizar el proceso en un espacio de tiempo razonable y poder además identificar “las agujas en el pajar”. En contraste, métodos de simulación con un grado de varios órdenes de precisión mayor tales como la Dinámica Molecular (*DM*) [22], [23] y la teoría de perturbación de la energía libre, requieren actualmente de cientos a miles de horas de CPU para poder procesar cada ligando [24]–[26]. Los métodos de CV deben utilizar una serie de aproximaciones, que en algunos casos conducen a resultados erróneos, para poder realizar la estimación de la afinidad de

unión del ligando a la proteína en cuestión de minutos o como máximo horas por ligando.

La caracterización experimental en los laboratorios y la optimización de estos compuestos son métodos costosos y lentos [27] pero la bioinformática puede ayudar enormemente en la investigación clínica para los fines mencionados al proporcionar la predicción de la toxicidad de los fármacos y la actividad en los objetivos no probados, y avanzar en el descubrimiento de compuestos activos en fármacos para los ensayos clínicos.

1.1.2 INTELIGENCIA COMPUTACIONAL

De un tiempo a esta parte, muchas de las técnicas desarrolladas en la estadística clásica, así como en la inteligencia artificial han sido puestas en práctica en un intento de construir modelos de predicción de comportamientos de forma automática y bajo una base estadística bien fundamentada

La búsqueda de patrones útiles se conoce con diferentes términos en diferentes comunidades (extracción de conocimiento, descubrimiento de información, procesamiento de patrones de datos). Este es un proceso no trivial de identificar patrones válidos, nuevos, potencialmente útiles, y comprensibles a partir de datos [28].

Los algoritmos de Reconocimiento de Patrones [29], son una disciplina que hace tiempo que salió de los laboratorios y las publicaciones científicas para impregnar nuestro día a día. Sistemas que reconocen la escritura [30], la voz [31], las imágenes [32], que descifran los genes [33], diagnostican enfermedades [34], interpretan las señales de trafico [35] o rechazan el correo basura [36]. Todos ellos, son algunos ejemplos

de estos sistemas con los que de manera casi imperceptible nos hemos acostumbrado, poco a poco, a convivir.

En las últimas décadas, comunidades científicas como, las de la estadística clásica, el reconocimiento de patrones [37], la comunidad de la inteligencia artificial o el aprendizaje automático, han extendido sus áreas de aplicación de forma notoria, aumentando la capacidad de extraer valiosos conocimientos de las grandes bases de datos información de distintos tipos desarrollando multitud de modelos predictivos explicativos [28].

La mayoría de los algoritmos de extracción de datos [38], se pueden ver como una combinación de unas pocas técnicas y principios y tienen en común tres componentes básicos:

- *El modelo*: este componente principal tiene dos factores relevantes: su función (clasificar, agrupar, resumir..), y el modo de representar el conocimiento (una función lineal de múltiples variables, un árbol, conjunto de reglas, una red..). Un modelo contiene ciertos parámetros que deben determinarse a partir de los datos.
- *El criterio de preferencia*: es la base para escoger un modelo o un conjunto de parámetros sobre otros. El criterio suele ser una función que hace que el modelo se ajuste a los datos que se disponen.
- *El algoritmo de búsqueda*: La especificación de un algoritmo para obtener modelos particulares y parámetros, los datos, el modelo (o familia de modelos), y un criterio de preferencia.

Las funciones más comunes de estos modelos incluyen:

- *Clasificación*: un clasificador es una función que asigna a una muestra no etiquetada una etiqueta o clase. Se clasifica un

caso entre varias clases o categorías predefinidas. Los modelos de clasificación se pueden construir utilizando una gran variedad de algoritmos [39] .

- *Regresión*: clasifica un caso con una variable de predicción de valor-real. En la regresión se persigue la obtención de un modelo que permita predecir el valor numérico de alguna variable [40].
- *Clustering (agrupamiento)*: clasifica un caso en una de las clases o agrupaciones en las que las clases se deben determinar a partir de los propios datos. Los clústers se definen buscando agrupaciones naturales de tos datos basado en modelos de medidas de similaridad, densidad de probabilidad o distancia [41].
- *Sumarización (resumen)*: provee una descripción compacta de un subconjunto de datos de entrada (media y desviación estándar para todos los campos, o reglas de resumen, relaciones funcionales entre variables) [42].
- *Modelado de dependencias*: describe las dependencias significativas entre variables. Existen modelos de dependencias a dos niveles: el estructurado y el cuantitativo. El modelo estructurado de dependencias especifica (a menudo en modo gráfico) qué variables son localmente dependientes: el modelo cuantitativo especifica la fortaleza de las dependencias usando una escala numérica [43].
- *Análisis de secuencias*: modela patrones secuenciales (como datos con dependencia temporal). El objetivo es modelar los estados del proceso generando la secuencia, o extraer y describir desviaciones y tendencias sobre el tiempo [44].

Para los métodos de inteligencia computacional un clasificador no es más que una función que dado un vector de características C asigna a este una etiqueta y lo hace perteneciente al conjunto de una clase a predecir. El entrenamiento de los métodos y la predicción final seguirá un esquema de combinación propias del método empleado.

Entre los métodos de inteligencia computacional es importante repasar tales conceptos, así como los trabajos realizados con redes neuronales o maquinas de soporte vectorial por sus capacidades de clasificación y los bosques aleatorios por su eficiencia en la selección de variables con mayor importancia o poder clasificador.

1.1.2.1 APRENDIZAJE

Una característica principal dentro de la inteligencia computacional es el paradigma de aprendizaje de los sistemas. Existen diversas definiciones de aprendizaje automático, entre ellas: "aprendizaje denota cambios en el en el sistema que son adaptativos en el sentido de que permiten al sistema realizar una misma tarea y la próxima vez lo harán de una forma mas eficiente y efectiva" [45]. Forsysth [46] especificó que, "el aprendizaje es un fenómeno que se muestra cuando un sistema mejora su rendimiento en una determinada tarea sin necesidad de ser reprogramado". En 1991 Weiss y Kulikowski [47] lo explicaron como: "un sistema que aprende es un programa de computador que toma decisiones en base a la experiencia acumulada contenida en casos resueltos satisfactoriamente". A diferencia de los sistemas expertos, que resuelven los problemas utilizando un modelo por computador del razonamiento de un experto humano, un sistema de aprendizaje puro puede utilizar muchas técnicas diferentes para explotar el potencial del computador, sin importar su relación con el proceso cognitivo humano. Para Langley

[48], "aprendizaje es la mejora en el rendimiento en ciertos entornos por medio de la adquisición de conocimiento como resultado de la experiencia en dicho entorno". Aunque el despegue del aprendizaje automático se produce en los años ochenta, la búsqueda de sistemas con capacidad de aprender se remonta a los primeros días de los computadores.

La adquisición del conocimiento por parte de los sistemas de aprendizaje automático se puede realizar de diferentes formas, igual que ocurre en los humanos que no tienen una única forma de aprender, aunque todos los paradigmas de aprendizaje se pueden encuadrar en las definiciones antes enunciadas, ya que todos tienen como objetivo común el incremento del rendimiento del sistema que adquiere el conocimiento.

Dentro del paradigma automático, nos encontramos con el aprendizaje supervisado, el cual genera hipótesis utilizando ejemplos con etiqueta (clase) conocida. A su vez, dichas hipótesis servirán para hacer predicciones ante nuevos ejemplos con etiqueta desconocida [49]. Dentro de un marco más operativo, el objetivo del aprendizaje supervisado (tanto binario como multiclase) es dividir el espacio de instancias (ejemplos) en regiones en donde la mayoría de los casos están etiquetados con la misma clase: dicha partición es la que servirá para predecir la clase de nuevos ejemplos. Al sistema se le proporciona un conjunto de hechos etiquetados y el sistema debe obtener el conjunto de reglas que expliquen estos hechos.

Uno de los problemas más antiguos de la investigación en este campo es encontrar funciones que ajusten, o expliquen, los datos que se observan en los fenómenos naturales [50]. La principal ventaja de la existencia de tales funciones es la posibilidad de predecir el comportamiento del sistema naturales en el futuro y controlar sus salidas mediante la aplicación de las entradas adecuadas. Algunos ejemplos interesantes podrían ser la predicción de valores en bolsa, la predicción meteorológica

o la clasificación de formas tumorales. La dificultad estriba en que los datos observados tienden a ir acompañados de ruido, y los mecanismos exactos que los generan normalmente son desconocidos. En ocasiones será posible encontrar un modelo matemático exacto que explique el proceso del que provienen los datos que observamos. Muchas veces, sin embargo, no podremos dar detalles de ese proceso. El objetivo, en este caso, será estimar el modelo subyacente que genera los datos observados.

1.1.2.2 REDES NEURONALES

Una red de neuronas artificiales (*RNA*) es un paradigma de procesamiento de información inicialmente inspirado en el modo en el que lo hace el cerebro, elemento clave de este paradigma es su estructura. Las RNA están compuestas por en cierto número de elementos de procesamiento o neuronas que trabajan al unísono.

A nivel histórico, se llevan estudiando desde la década de los 50, pero la Red de Hopfield, supuso el resurgimiento del campo de las redes neuronales tras la dura crítica impuesta por Minsky y Papert [51]. Los principales usos de esta red son como memoria asociativa y como herramienta para la resolución de problemas de optimización. Las redes neuronales artificiales tratan de emular tres conceptos claves:

- Procesamiento paralelo: derivado de que los miles de millones de neuronas que intervienen, por ejemplo en la acción humana de ver un objeto, es completamente paralela, y se realiza sobre toda imagen a la vez.
- Memoria distribuida: en las redes neuronales biológicas la información esta distribuida por las sinapsis de la red, existiendo una redundancia en caso de que una sinapsis resulte dañada,

mientras que en un computador la información está en posiciones de memoria.

- Adaptabilidad al entorno: la información de las sinapsis se adapta y esta adaptabilidad hace que se puede aprender de la experiencia y sea posible generalizar conceptos a partir de casos particulares

Aunque no existe una definición general de red neuronal artificial, y existen diferentes versiones, según el texto o artículo consultado. Así, podemos citar algunas de estas definiciones:

- Una red neuronal es un modelo computacional, paralelo, compuesto de unidades procesadoras adaptativas con una alta interconexión entre ellas [52].
- Son sistemas de procesado de la información que hacen uso de algunos de los principios que organizan la estructura del cerebro humano [53].
- Son modelos matemáticos desarrollados para emular el cerebro humano [54].
- Es un sistema de procesado de la información que tiene características de funcionamiento comunes con las redes neuronales biológicas [55].
- Sistema caracterizado por una red adaptativa combinada con técnicas de procesado paralelo de la información [56].
- Desde la perspectiva del reconocimiento de patrones, las redes neuronales son una extensión de métodos clásicos estadísticos [57].

La arquitectura de una red neuronal es la topología, estructura o patrón de conexionado de sus elementos (ver Figura 1.1). En una red neuronal artificial los nodos o elementos se conectan por medio de sinapsis, estas

conexiones sinápticas determinadas por la estructura que conforman su comportamiento.

Las conexiones sinápticas son direccionales, el sentido en que la información se propaga en un único (desde la neurona presináptica a la pos-sináptica). Las neuronas se agrupan por lo general en unidades estructurales, capas. La red neuronal constituye el conjunto de una o más capas.

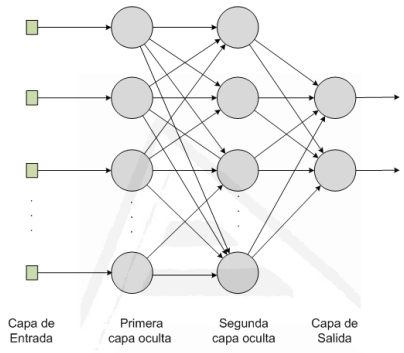


Figura 1.1: Esquema de Red Neuronal

Existen tres tipos de capas: de entrada, de salida y ocultas.

- La capa de entrada, es la sensorial, aquí se reciben los datos o señales procedentes del entorno, y está compuesta por neuronas.
- La capa de salida, son las neuronas que proporcionan la respuesta de red neuronal.
- La capa oculta, sin conexión directa con el entorno, es capaz de representar las características del entono que modela.

Teniendo en cuenta la estructura podemos hablar de redes compuestas por una única capa o "redes monocapa", y cuando las neuronas se organizan en varias capas hablamos de "redes multicapa".

Teniendo en cuenta el flujo de datos, distinguimos entre redes unidireccionales (*feedforward*) donde la información circula en un único sentido y redes recurrentes o realimentadas (f*eedback*) en las que la información puede circular entre las distintas capas de neuronas en cualquier sentido, incluso en el de salida-entrada.

1.1.2.2.1 EL PERCEPTRÓN MULTICAPA

El perceptrón es quizás la forma más simple de una red neuronal que se puede utilizar para la clasificación de clases o conceptos que sean linealmente separables, es decir que las muestras positivas y negativas de la clase se pueden separar mediante un hiperplano en el espacio de características *X*, en las Fig. 1.2 y 1.3 se muestra un ejemplo para dimensión 2.

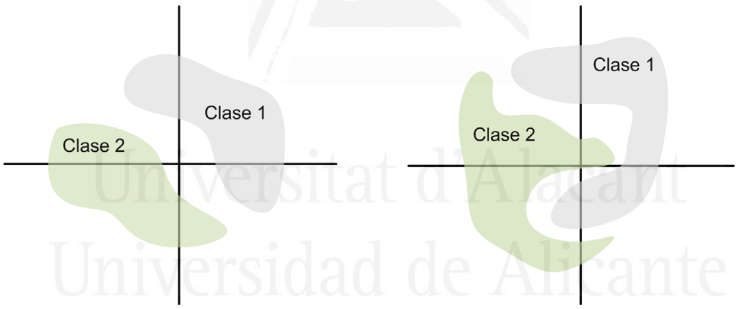


Figura 1.2: Clase linealmente separable. **Figura 1.3:** Clase no linealmente separable.

En una red neuronal es necesario definir un procedimiento por el cual las conexiones del dispositivo varíen para proporcionar la salida deseada. Se observa la salida de la red y se determina la diferencia entre ésta y la señal deseada. Posteriormente, los pesos de la red son modificados de acuerdo con el error cometido (algoritmo de aprendizaje).

Hace ya más de dos décadas, que se estudió la predicción de la solubilidad en agua de los compuestos orgánicos utilizando enfoques neuronales [58]. En la siguiente década, se emplearon modelos neuronales supervisadas y no supervisadas para modelar QSAR (*Quantitative Structure–Activity Relationship*) [59], predecir actividades y estructuras moleculares, la agrupación de estas y muchos otros [60], [61] . Más recientemente, el problema de la predicción de la solubilidad de los fármacos basados en su estructura molecular ha sido revisado [62]. La predicción de las propiedades físico-químicas de los compuestos orgánicos a partir de la estructura molecular ha sido ampliamente estudiado mediante el uso de técnicas híbridas que incluyen redes neuronales [63]–[65]. También la identificación de ligandos de pequeñas moléculas ha mejorado con el uso de técnicas neuronales [66]–[68].

Podemos encontrar aplicaciones a las redes neuronales en muchos otros campos de aplicación:

- Medicina, diagnóstico de cardiopatías [69], decisión en desfibriladores [70], compresión de señales electrocardiográficas [71], predicción de enfermedades degenerativas cardíacas [72].
- Farmacia, efectos adversos de la administración de un fármaco [73], predicción de la respuesta emética (número de naúseas y vómitos de un paciente oncológico) [74].
- Procesado de la señal, ecualización de canales de comunicación (recuperaración de la señal que sufre distorsión) [75], cancelación activa de ruido [76].
- Reconocimiento de patrones, imágenes [77], reconocimiento de voz [78], sónar y rádar [79].
- Economía, predicción del gasto [80], la falta o un exceso de suministros [81].

- Medio Ambiente, la predicción de irradiación solar [82] y la predicción de variaciones globales de temperatura [83].

1.1.2.3 MÁQUINAS DE SOPORTE VECTORIAL

Las Máquinas de Soporte Vectorial (*SVM, Support Vector Machine*) [84] son un grupo de métodos de aprendizaje supervisado que se pueden aplicar a la clasificación o regresión. Representan la frontera de decisión en términos de un típico pequeño subconjunto de todos los ejemplos de entrenamiento, llamado los vectores de soporte.

Una SVM es un clasificador lineal en un espacio que podría ser distinto al espacio original donde están definidos los vectores X , y por tanto un hiperplano que clasifica las instancias por la pertenencia a cada una de las regiones de ese espacio que son limitadas por dicho hiperplano. Las SVM tienen como objetivo encontrar el hiperplano óptimo que separe las dos clases y maximice el margen. Las SVM dividen el espacio del problema en dos, por tanto son clasificadores binarios. Si bien existen técnicas basadas en generar múltiples SVM para tratar el caso multiclase.

Podemos encontrar investigaciones sobre las relaciones cuantitativas estructura-actividad (*QSAR*), cuando se utiliza la regresión SVM para predecir diversas propiedades químicas, biológicos o físicas [85], la quimiometría (optimización de la separación cromatografía o predicción de la concentración del compuesto a partir de datos espectrales como ejemplos), sensores (para la predicción cualitativa y cuantitativa de los datos de sensores), la ingeniería química (detección de fallos y la modelización de procesos industriales) [86]. Una excelente revisión de las aplicaciones de SVM en la química fue publicado por Ivancicuc [87].

Algunos ejemplos de disciplinas que se han sumando al uso de Maquinas de Soporte Vectorial son:

- Ingeniería mecánica [88].
- Ingeniería Financiera [89], [90], [91], [92].
- Modelos de Lenguaje, [93], [94].
- Medicina [72], [95], [96] y Biología [97].
- Reconocimiento de Patrones, de Escritura [98] y Facial [99], [100].
- Seguridad en Internet [101], [102].
- Data Mining [103], [104].

1.1.2.4 BOSQUES ALEATORIOS

Bosques Aleatorios (RF, Random Forest) [105] es un método de aprendizaje supervisado que se puede aplicar a la clasificación o regresión, mediante una combinación de árboles predictores.

Random Forest es "una colección de clasificadores estructurados como árboles t_n donde $F_n(v)$ son vectores aleatorios independientes e idénticamente distribuidos, y cada árbol produce un voto de la clase más popular para una entrada x (predictor)". Los vectores aleatorios $P_n(c)$ representan un conjunto de números aleatorios que determina la construcción de cada árbol (ver Figura 1.8).

La implementación más sencilla y común, consiste en que para cada árbol compuesto de nodos, estos solo se pueden ramificar a partir de un subconjunto del conjunto de atributos (predictores) de partida. Este subconjunto es distinto para cada nodo y aleatorio en cuanto a su composición. El tamaño de los subconjuntos es fijo y se especifica como parámetro dentro del entrenamiento de partida.

En poco tiempo multitud de disciplinas se han sumando al uso de Random Forest en campos tales como:

- Análisis de accidentes [106].
- Ingeniería mecánica [107].
- Ingeniería financiera [108], [109].
- Modelos de lenguaje [110].
- Biología [111] y química [112], [113] y [114].
- Data mining [115].
- Seguridad de redes [116].
- Geología [117].

1.1.3 ENTORNOS DE COMPUTACIÓN DE ALTO RENDIMIENTO

Gracias a la alta escala de integración que permite la tecnología VLSI (*Very Large Scale Integration*), desde finales de la década de los años 90 la comunidad investigadora se ha planteado cómo organizar los chips para hacer un uso lo más eficiente posible de la ingente cantidad de transistores de que se va a disponer a corto y medio plazo.

Una importante cuestión a la hora de implementar un microprocesador tradicional con una elevada cantidad de transistores es la complejidad de diseño que presenta. No es nada sencillo diseñar un procesador superescalar normal, escalado para utilizar eficientemente miles de millones de transistores y menos sencillo todavía validar su correcto funcionamiento. A pesar de la mejora en las herramientas de diseño y del aumento del número de diseñadores involucrados en una propuesta concreta, la dificultad del diseño es tal que es difícil realizar su validación en el tiempo requerido por las restricciones impuestas por el mercado, por lo que se relentiza su evolución.

El resultado de estas tendencias es que el diseño de un procesador formado por miles de millones de transistores está organizado en pequeños y localizados elementos de procesamiento, de tal forma que los recursos que deban comunicarse entre sí, dentro del chip del procesador, estén físicamente cercanos. Esto ha dado lugar a las arquitecturas multicore o CMP (*Chip Multiprocesor*). Los procesadores CMP están formados por procesadores que ejecutan habitualmente un sólo flujo de instrucciones con un nivel de paralelismo interno moderado, permitiendo también la ejecución de múltiples hilos en paralelo por medio de múltiples cores. La alternativa de los multicores fue consolidada en 2005, cuando Intel siguió la dirección de los procesadores IBM Power 4 y Sun Microsystem Niagara anunciando que sus procesadores mejorarían su rendimiento ampliando el número de elementos de cómputo (núcleos o cores) dentro del chip. Estos cores siguen siendo procesadores complejos, es decir, son diseñados para acelerar al máximo la ejecución de programas secuenciales (procesadores fuera de orden, implementación de todo el juego de instrucciones x86, etc...). Esta tendencia en el diseño de los multicores ha sido objeto de importantes esfuerzos de investigación, se han consolidado diversas propuestas tanto comerciales (IBM Power4, IBM Power5, Cell) como de investigación (Piranha, Hydra o TRIPS).

1.1.3.1 UNIDADES DE PROCESAMIENTO GRÁFICO PARA PROPÓSITO GENERAL

En el año 2002, Mark Harris bautizó el movimiento de investigación que utilizaba la GPU para procesamiento de aplicaciones no gráficas como GPGPU (*General-Purpose Computation on Graphics Hardware*) o GPU Computing. La GPU se empezó a ver como una alternativa de altas prestaciones "manycore", es decir, que contenía una gran cantidad de núcleos o cores dentro del chip. Esta gran cantidad de cores era posible

gracias a la reducción de la complejidad de los mismos. Al igual que en el caso de los multicore, en los manycore el número de cores se dobla en cada generación de semiconductores. Uno de los primeros ejemplos fue la unidad de procesamiento gráfico de NVIDIA GeForce GTX 280 que contenía hasta 240 cores. Los manycores, y concretamente las GPUs han liderado la carrera del rendimiento en punto flotante (*FLOPS, floatingpoint operations per second*) desde 2003. Esta tendencia en el diseño de manycores también está siendo objeto de importantes esfuerzos de investigación, y las grandes empresas del sector están apostando por esta alternativa (Intel Larrabee, AMD/ATI Firestream technology).

El principal problema de esta tecnología hasta el año 2006 era la ardua tarea que suponía la programación de aplicaciones no gráficas en la GPU. Los programadores tenían que lidiar con las interfaces de programación gráficas (API), tales como DirectX o OpenGL, para acceder a los cores de la GPU. La necesidad de usar estas APIs gráficas limitaba el tipo de aplicaciones que los programadores podrían desarrollar para estos chips. NVIDIA dedicó parte de los transistores de la GPU para facilitar la programación paralela de aplicaciones de propósito general a partir de la arquitectura G80, y además creó un lenguaje de programación (*CUDA, Compute Unified Device Architecture*) basado en C/C++, mucho más sencillo y flexible que el tradicional API gráfico. La aparición de CUDA [118] de NVIDIA cambió el panorama de la GPGPU.

1.3 PROPUESTA DE SOLUCIÓN

Los métodos de CV contienen una serie de deficiencias tanto a nivel de precisión como de velocidad de cómputo, y esto es un cuello de botella para poder descubrir nuevos compuestos bioactivos o mejorar los ya existentes. Se propone el uso de métodos de inteligencia computacional basados en redes neuronales (NNET), maquina de soporte vectorial (SVM) y bosques aleatorios (RF), para refinar la predicción de candidatos y que mejoren ostensiblemente los métodos de CV.

El desarrollo de una metodología que refine las predicciones y ayude en la optimización del descubrimiento de compuestos bioactivos y su aplicación a problemas de relevancia biomédica. Para ello proponemos utilizar técnicas de inteligencia computacional para incrementar la precisión de los métodos de CV sobre arquitecturas paralelas de altas prestaciones y bajo coste.

La capacidad predictiva de los métodos de CV se ha estancado en la última década, para poder obtener mejoras en métodos CV hay que refinar la predicción de candidatos, esto posibilitará el diseño de aplicaciones biomédicas que sean más eficientes tanto en tiempo como en energía consumida, y además sean económicamente rentables, y permita incrementar el grado de realismo en los modelos biomédicos utilizados.

Los cuellos de botella presentes en las metodologías de CV condicionan el uso de la aceleración en GPUs; los cálculos requeridos por los métodos de CV son computacionalmente muy costosos, mucho más todavía cuando crece la cantidad de compuestos químicos a estudiar. El acceso a recursos de supercomputación permite resolver este problema, pero esta solución no es práctica para toda la comunidad científica, dado el gran desembolso económico que implica. La explotación de GPUs para poder

realizar los cálculos necesarios permitiría acelerarlos de manera drástica, por un presupuesto muchísimo menor y a un consumo eléctrico mucho más reducido.

Para que los métodos de CV sean capaces de procesar millones de ligandos en poco tiempo, es necesario que estos recurran a ciertas simplificaciones, el refinamiento que proponemos con la introducción de la predicción de actividad para los grandes dataset, permitirá el empleo otras técnicas de simulación molecular más precisas al disminuir el número de compuestos con los que probar.

La mejora de las predicciones de afinidad de las interacciones proteína-ligando mediante la explotación de resultados experimentales previos; que pueden ser potencialmente bioactivos y que más tarde son caracterizados y la información obtenida experimentalmente se aprovecha de manera adecuada para proporcionar *feedback* a los métodos de CV de tal manera que sucesivas etapas de refinamiento puedan incrementar su capacidad predictiva. Proponemos por tanto en esta parte desarrollar y acoplar a métodos de CV una estrategia que explote esta información experimental de manera eficiente.

El resumen de la propuesta se presenta a modo de gráfico en el esquema de la figura 1.4. La mejora de la predicción de actividad mediante técnicas de inteligencia computacional suponen un refinamiento del método de CV BINDSURF de forma que utilizamos esta ganancia para mejorar la predicción de la afinidad ligando-proteína; a) el método Virtual BINDSURF, y b) se estudian dos técnicas de inteligencia computacional ; redes neuronales (NNET), máquinas de soporte vectorial (SVM) y bosques aleatorio (RF), entrenadas con diferentes propiedades moleculares calculados para compuestos conocidos activos e inactivos seleccionados de conjuntos de datos referencias estándar para CV. En la Figura 1.4 se muestra un diagrama de flujo de la metodología; una vez que se han

elegido un objetivo para la proteína (componente A) y una base de datos de compuesto (componente B), los compuestos para los cuales se dispone de información acerca de la afinidad contra la proteína objetivo (componente C) se acoplan mediante BINDSURF (componente D) y las afinidades estimadas (componente E se obtienen poses) y 3D (Componente F).

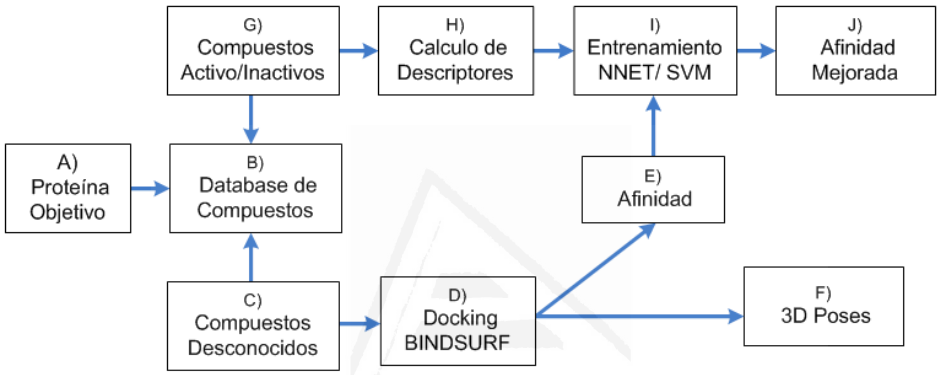


Figura 1.4. Diagrama de flujo de la metodología usada para el refinamiento de la capacidad predictiva de BINDSURF.

Utilizando los métodos descritos en esta sección, comenzamos seleccionando compuestos de la base de datos para la que existen datos disponibles de afinidad (componente G), de modo que podamos calcular los descriptores relevantes (componente H) y entrenar de manera adecuada las redes neuronales y las máquinas de soporte vectorial (componente I) por lo que las afinidades obtenidos en el componente E son postprocesadas y finalmente obtienen valores mejorados para las afinidades (componente J) .

1.4 METODOLOGÍA

Este trabajo fruto es de la investigación multidisciplinar y aúna conocimientos procedentes de diferentes campos de la ciencia, principalmente de la computación y de la química. Para su desarrollo ha sido necesario el uso de herramientas y métodos de diversa índole que se muestran a continuación.

1.4.1 MÉTODOS COMPUTACIONALES PARA EL DESCUBRIMIENTO DE FÁRMACOS

La actividad de muchas proteínas cambia de manera drástica cuando pequeñas moléculas denominadas ligandos se acoplan a determinadas zonas de los receptores proteicos. Estos ligandos pueden actuar como interruptores moleculares de las proteínas y regular su actividad. En el caso de proteínas implicadas en rutas metabólicas relacionadas con una enfermedad, ciertos ligandos, ya sean naturales o diseñados artificialmente, pueden actuar como fármacos.

Actualmente existen diversos métodos de CV tales como AutoDOCK [4], FlexScreen [119], y BINDSURF [120]. Estos métodos permiten una exploración cuasi-exhaustiva de las diferentes conformaciones posibles que puede adoptar el ligando durante el proceso de acoplamiento al receptor, prediciendo correctamente en la mayoría de los casos la conformación experimental de unión aunque por otra parte, suelen presentar escasa correlación entre la afinidad de unión predicha y el valor experimental [121] siendo por tanto, complicado obtener el conjunto completo de compuestos de alta afinidad a partir de una gran librería de ligandos. La solución que nosotros proponemos para este problema consiste en refinar los métodos de predicción de actividad

utilizando técnicas de inteligencia computacional, con el fin de permitir posteriormente utilizar nuevas estrategias para simular el proceso de acoplamiento del ligando en el receptor, esto permitirá a los métodos de CV acelerar la velocidad a la que se realizan los cálculos con detalle atómico e incrementar y mejorar el nivel de realismo con que se evalúa la intensidad de interacción receptor-ligando, al tener que evaluar un menor número moléculas.

Posteriormente se realizan estudios in-vitro e in-vivo para comprobar la actividad real de dichas moléculas y avanzar a fases siguientes de optimización del compuesto (eliminar toxicidad, incrementar potencia y eficacia), así como realizar ensayos clínicos en pacientes humanos.

1.4.1.1 ACOPLAMIENTO MOLECULAR

Una de las técnicas más utilizadas para evaluar las interacciones entre compuestos con actividad promiscua biológica y un receptor es el acoplamiento (*docking*) molecular, debido a que la función principal de los fármacos es enlazarse en algún punto activo del receptor. El *docking* molecular consiste en calcular la energía de interacción entre las dos moléculas, receptor y ligando, en función de la energía libre de Gibbs [122]. Dado el gran tamaño de los sistemas macromoleculares, así como el gran número de posibles conformaciones del sistema y posibles interacciones del ligando con el receptor, está claro que es necesario utilizar un modelo que evalúe con eficiencia y reproducibilidad las diferentes interacciones sustrato-sitio activo. La mecánica molecular que describe los sistemas mediante campos de fuerza, interacciones enlazantes como: enlaces, torsiones, ángulos e interacciones no enlazantes como: puentes de hidrógeno, Van der Walls y electrostáticas, que están basados en métodos clásicos. Los campos de fuerza (AMBER

[123], CHARMM, GROMOS, UFF, MM4) se han desarrollado con base a datos espectroscópicos, datos experimentales y cálculos iniciales, razón por la cual en muchos casos se ha demostrado que brindan respuestas formalmente correctas, a pesar de que no incluyen en la descripción de las propiedades moleculares el movimiento electrónico (poses).

Los métodos de docking son los más usados en el CV, y proporcionan predicciones de las conformaciones finales de los complejos proteína-ligando así como de las afinidades de la unión. La precisión de sus predicciones se ha estancado en los últimos diez años [124].

1.4.1.2 DESCRIPTORES MOLECULARES

El tratamiento de la información y el conocimiento de modelos a partir de datos del mundo real hace que sea necesario definir las propiedades que diferencian a ciertos objetos de otros. Es necesaria una definición explícita de una descripción formal, de forma que se conserva la distinción natural entre objetos. Es obvio que la forma de un objeto describe y depende del contexto y del dominio de interés. En el caso de estructuras moleculares, la descripción elegida del mismo compuesto ciertamente sería diferente, si se describe como una afinidad específica como una diana farmacéutica o como su síntesis experimental. Por esta razón, se han propuesto literalmente miles de descriptores moleculares que cubre todas las propiedades de interés para múltiples dominios.

Un descriptor molecular en la mayoría de los casos, es un valor numérico asociado a la propiedad de una estructura molecular, derivado por algún algoritmo que describe un aspecto específico de un compuesto. Hay muchas maneras de definir las clases de los descriptores. El objeto más importante es diferenciar entre las representaciones estructurales utilizadas como entrada. Los tipos más simples son descriptores

unidimensionales (1D, 0D) que sólo dependen de la formula molecular, como la masa molecular o un número de elementos específicos. La carga neta de una molécula es a menudo considerado como un descriptor de 1D.

Nombre Descriptor	Elementos
Constitutional descriptors	30
Topological descriptors	35
Connectivity descriptors	44
Kappa descriptors	7
Basak descriptors	21
E-state descriptors	245
Burden descriptors	64
Autocorrelation descriptors	96
Charge descriptors	25
Molecular property descriptors	6
MOE-type descriptors	60
Geometric descriptors	12
CPSA descriptors	30
WHIM descriptors	70
MoRSE descriptors	210
RDF descriptors	180
Fragment/Fingerprint-based	8

Tabla 1.1. Diferentes grupos de descriptores moleculares.

La mayoría de los descriptores tienen en cuenta la topología molecular (es decir, la fórmula estructural). Estos son considerados como de dos

dimensiones (2D) como los descriptores basados en la teoría de grafos. Descriptores que también consideran la estructura espacial se definen en tres dimensiones (3D). Otras clases de descriptores que se han introducido, muestran diferentes conformaciones, su dimensionalidad no puede expresarse de una manera intuitiva por lo que se expresan como, de cuatro (4D) o de cinco dimensiones (5D). En la tabla 1.1 mostramos una relación de descriptores moleculares y número de elementos que los componenen.

1.4.1.3 DATASETS PROTEÍNA-LIGANDO

El uso de las diferentes conjuntos de bases de datos de moléculas bioactivas son una prueba estándar y de referencia para CV. Los dataset como el DUD (*Directory of Useful Decoys*) [125], se han comprobado que son eficientes diferenciando los ligandos que se unen a una molécula objetivo. Los datos de entrada para cada molécula de cada conjunto contiene información acerca de su estructura molecular. Nos hemos centrado en tres conjuntos de datos DUD (los detalles se muestran en la Tabla 1.1) diversos que cubren Kinasas, receptores nucleares de hormonas y otras enzimas. Estos DUD están codificadas con el Código PDB (*Protein Data Bank Code*) [1] y hemos utilizado los datasets, TK (PDB code 1KIM) que corresponden a Thymidine Kinasa, MR (PDB code 2AA2) que corresponde al receptor de Mineralocorticoides y GPB (PDB coe 1A8I) que corresponde a la Enzima Glucógeno Fosforilasa.

A continuación, utilizando el paquete ChemoPy [126] se calculó para todos los ligandos de los conjuntos de datos TK , MR y GPB las propiedades moleculares derivados del conjunto Constitucional , CPSA

(superficie parcial cargada) y los descriptores basados en fragmento de huella moleculares (FFP).

Proteína	PDB Code	Resolución (Å)	Ligands	Decoys
GPB	1A8I	1.8	52	1851
MR	2AA2	1.9	15	535
TK	1KIM	2.1	22	785

Tabla 1.2. Número de compuestos bioactivos (ligands) y los compuestos inactivos (decoys) para cada uno de los conjuntos de datos de ligandos usados en este estudio y obtenidos a partir de DUD (Directory of Useful Decoys).

1.4.2 ARQUITECTURAS PARALELAS DE ALTO RENDIMIENTO: GPUS

La escalabilidad de las aplicaciones para sistemas paralelos y la portabilidad entre sistemas de distinta naturaleza son también otros factores críticos para el rendimiento de las aplicaciones paralelas. Debido al continuo incremento de cores de los sistemas paralelos actuales, el objetivo principal subyacente es conseguir que la aplicación sea más rápida conforme aumente el grado de paralelismo del sistema, es decir, tenga una cierta escalabilidad entre nuevas generaciones de sistemas paralelos. Además, la portabilidad de las aplicaciones paralelas entre sistemas paralelos de distinta naturaleza es otro parámetro fundamental para el desarrollo de aplicaciones que mejoren el rendimiento de las mismas en el futuro.

1.4.2.1 RENDIMIENTO DE LAS APLICACIONES PARALELAS

De cara a obtener el mejor rendimiento posible de una aplicación paralela, un primer factor a tener en cuenta es identificar su patrón de cómputo y estudiar la idoneidad para cada tipo arquitectura. Uno de los mayores obstáculos para innovar en computación paralela es la falta de un mecanismo de descripción de los problemas en términos de paralelismo. Por tanto, existe una necesidad de encontrar un nivel más alto de abstracción para razonar sobre los requisitos de una aplicación paralela. Se ha creado un conjunto de benchmarks, llamados "dwarfs", que definen un patrón de cómputo y de comunicaciones comunes para un conjunto importante de aplicaciones y, en base a esos patrones, se describen las mejores plataformas para ejecutar dichas aplicaciones.

El rendimiento de cualquier aplicación, y en concreto de las aplicaciones paralelas, está estrechamente ligado al modelo de programación subyacente de la propia aplicación. Muchos modelos de programación paralelos han sido propuestos durante las últimas décadas. Los más usados son MPI (*Message Passing Interface*) para clúster de procesadores con un modelo de memoria distribuida, y OpenMP para multiprocesadores de memoria compartida. Estos dos modelos de programación son estándares de programación paralela, especialmente para aplicaciones codificadas según el paradigma SPMD (un único programa trabajando sobre múltiples datos). CUDA ofrece un nuevo modelo de programación desarrollado principalmente para un paradigma SIMD (una misma instrucción ejecutada sobre múltiples datos), aunque también permite utilizar el modelo SPMD para describir el problema a alto nivel, y se ha demostrado muy eficaz para codificar aplicaciones paralelas de propósito general en el entorno de las GPUs.

1.4.2.2 DESCUBRIMIENTO DE FÁRMACOS Y EXPLOTACIÓN DE GPUS

Para poder realizar los complejos y costosos cálculos requeridos en las simulaciones es necesario tener acceso a supercomputadores, lo cual es muy costoso y no es accesible a la mayoría de los investigadores. En definitiva, el uso de GPUs, puede solucionar de manera drástica el problema de la gran necesidad de cómputo que tienen los métodos de CV. Las GPUs han ganado últimamente mucha popularidad en el campo de la computación de alto rendimiento gracias a la combinación de su enorme potencial para realizar cálculos complejos, junto con los requerimientos de la industria de los gráficos por ordenador y de los videojuegos [9]. Algunos investigadores han comenzado a explotar este poder en muy diversos dominios computacionales, y finalmente las GPUs han emergido como un elemento clave en todas aquellas aplicaciones en las cuales el paralelismo es el denominador común. Por tanto, las GPUs están bien preparadas para solucionar el problema de la demanda de recursos computacionales de los métodos de CV, acelerando el tiempo de proceso requerido para sus cálculos [127].

La explotación de GPUs permitirá incluir en los métodos de CV ciertas características que anteriormente eran inviables, y ello permitirá incrementar el realismo y la calidad de predicción de estos métodos.

1.4.2.3 DOCKING EN GPUS: BINDSURF

Los métodos de cribado virtual realizan el descubrimiento de fármacos mediante el cribado de grandes librerías de compuestos químicos [128] . El programa de CV BINDSURF [120], encuentra ligandos que sean capaces de unirse a una proteína de estructura conocida, provenientes

de una librería o quimioteca que contiene las estructuras tridimensionales de dichos ligandos. BINDSURF simula el proceso de interacción de cada ligando por toda la superficie de la proteína (ver Figura 1.5), utilizando para ello una representación atómica de tanto la proteína como el ligando.

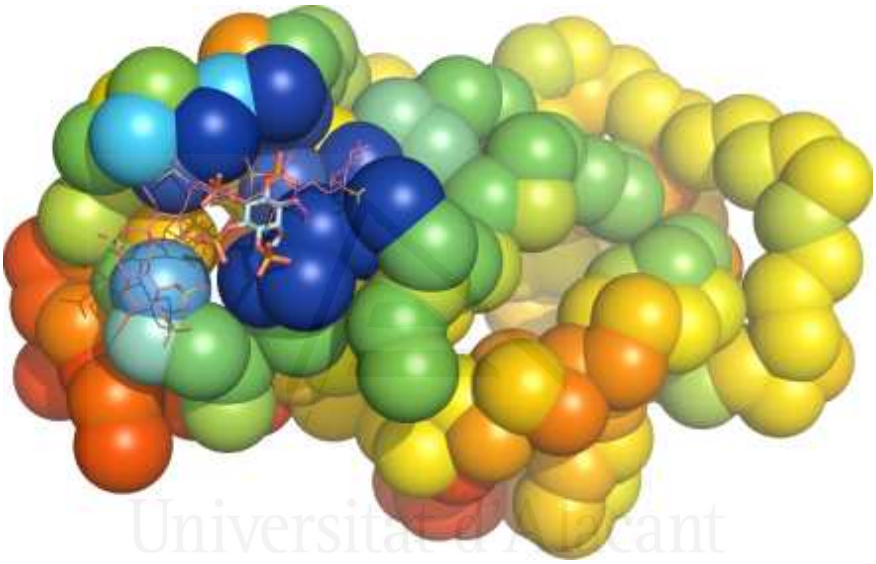


Figura. 1.5: Representación de los resultados de docking molecular de TMI (D-myo-inositol 3,4,5,6-tetrakisphosphate) obtenidos sobre toda la superficie de antitrombina. Cada simulación comienza en el centro de cada una de las bolas y el color de cada una de ellas representa la intensidad de interacción, yendo desde rojo (escasa interacción) a azul (fuerte interacción). El esqueleto de heparina figura en color rosa mientras que la conformación final predicha para TMI se resalta con un esqueleto de color azul claro.

BINDSURF no realiza ninguna suposición previa sobre la localización del sitio de unión del ligando en la superficie de la proteína, y esto le confiere una gran ventaja frente a otros métodos de CV.

La automatización parte de la detección de las zonas de interacción proteína-ligando; dependiendo de la estructura química de los ligandos, estos pueden interaccionar con diferentes partes de la proteína. Los métodos actuales de CV tienen una gran desventaja que consiste en usar siempre el mismo lugar de unión para todos los ligandos [129].

Después de una ejecución de BINDSURF, y con la información obtenida acerca de cómo interaccionan los diferentes ligandos sobre la superficie de la proteína, es posible entonces formular hipótesis que guíen la aplicación de otros métodos de CV más avanzados (pero computacionalmente mucho más costosos) tales como dinámica molecular [130] .

BINDSURF es una metodología muy eficiente para la determinación de sitios de unión de la proteína para los diferentes ligandos (en la Figura 1.6 se pueden ver la predicción de unión de dos ligandos diferentes sobre una misma proteína). Se puede utilizar para realizar el pre-cribado de grandes quimiotecas, con millones de compuestos químicos, y luego aplicar otras metodologías de CV más avanzadas tales como dinámica molecular. BINDSURF es capaz de obtener resultados en un tiempo muy corto, del orden de dos minutos por par proteína-ligando, y reproducir resultados experimentales obtenidos para una gran cantidad de complejos cristalográficos resueltos experimentalmente.

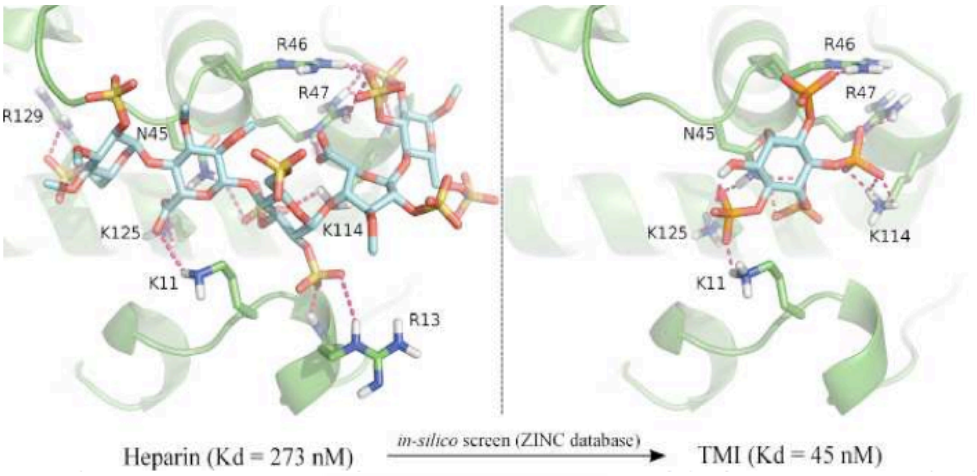


Figura 1.6: Predicción de unión para la heparina y D-myo-inositol 3,4,5,6-tetrakisphosphate (TMI), sobre una misma proteína.

A finalizar este 2014 no ha sido reportada la existencia de otro método de CV con tales características, capaz de realizar el cribado a semejante velocidad, otros métodos previos (que no usan GPUs) pueden resultar unas 100 veces más lentos. Se pretende la mejorar de las predicciones de los métodos de cribado virtual mediante el uso de técnicas de inteligencia artificial [131]

1.4.3 EL LENGUAJE DE PROGRAMACIÓN R

R es un lenguaje de programación [132] y un entorno para computación y gráficos estadísticos. Tanto el lenguaje R como su entorno (http://cran.r-project.org) han sido utilizados ampliamente para la realización de esta tesis, y son la base de toda la experimentación aquí expuesta. Se han utilizado múltiples librerías para conseguir satisfacer la experimentación

requerida (http://cran.r-project.org/web/packages) . El repositorio CRAN [133] a finales de este 2014, dispone de más de 6.000 paquetes.

R es un proyecto GNU [134], similar al lenguaje S (que fue desarrollado en los Laboratorios Bell por John Chambers) [135]. R puede ser considerada como una implementación diferente de S, hay algunas diferencias importantes, pero mucho código escrito para S corre inalterado bajo R, que fue desarrollado inicialmente por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland en 1993. Su desarrollo actual es responsabilidad del R *Development Core Team*.

R ofrece una amplia variedad de técnicas gráficas estadística (análisis de series de tiempo lineal y modelado no lineal, pruebas estadísticas clásicas, clasificación, agrupamiento, ...) y es altamente extensible.

R está disponible como software libre bajo los términos de la Licencia Pública General (*Free Software Foundation)* en forma de código fuente. Se compila y se ejecuta en una amplia variedad de plataformas UNIX , FreeBSD, Linux, Windows y MacOS.

1.4.4 MÉTODOS DE INTELIGENCIA COMPUTACIONAL

Desde la década de los 50 la investigación en inteligencia computacional se ha centrado en la búsqueda de relaciones entre los datos y análisis de extracción de tales relaciones. Estos problemas son encontrados en una amplia variedad de dominios de aplicaciones, ingeniería, robótica, reconocimiento de patrones (voz, escritura y reconocimiento facial), internet, medicina y bioinformática.

Dado un número datos de entrenamiento (training) asociados con una salida esperada, los procesos de inteligencia computacional nos

permitirán encontrar la relación entre el patrón y el resultado esperado, usando estos datos de entrenamiento. El objetivo es predecir la salida desconocida, para un conjunto de nuevo de datos (test). La generalización de esta tarea y la construcción de modelo predictivo o predictor, que contiene unos paramentos ajustables. Los datos de entrenamiento son utilizados para la selección óptima de esos parámetros, y los diferentes algoritmo a emplear constituyen un amplio abanico de técnicas de inteligencia computacional como redes neuronales (NNET), maquinas de soporte vectorial (SVM) y bosques aleatorios (RF)

1.4.4.1 EL PERCEPTRÓN MULTICAPA

Una de las áreas de aplicación mas importante de las redes neuronales es la aproximación de funciones no lineales. La principal ventaja de modelo de la red neuronal es que la complejidad de las relaciones no lineales puede se modelada sin suposiciones previas acerca de la forma del modelo. Esta característica es muy útil en el campo del diseño y descubrimiento de fármacos.

En los últimos años un gran número de autores han diseñado métodos híbridos que combinan redes neuronales con otras técnicas para resolver problemas relacionados con la química.

Hay varios tipos de redes neuronales con alimentación hacia adelante (NNET), las más ampliamente utilizadas son las multi-capa con función de activación sigmoidal (perceptrones multicapa) y las redes de una sola capa con funciones de activación local (redes de funciones de base radial). La buena capacidad de aproximación de las redes neuronales ha sido ampliamente demostrada para las aplicaciones prácticas y la investigación teórica. Hemos decidido utilizar una red neuronal de una capa oculta con solo conexiones entre etapas para este estudio (Figura

1.7) ya que se ha demostrado claramente su impacto en la clasificación entre compuestos activos e inactivos y otras aplicaciones químicas [60]. Para tal fin se utilizó la función NNET del paquete R [136] .

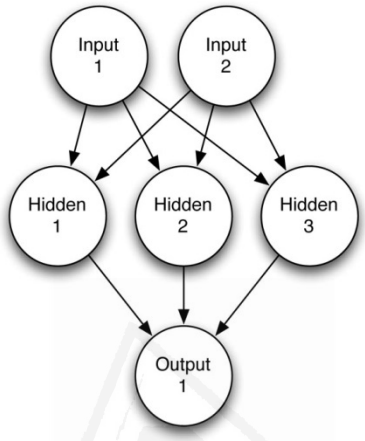


Figura 1.7. Red Neuronal de una sola capa oculta.

1.4.4.2 MAQUINAS DE SOPORTE VECTORIAL

En nuestro caso, explotamos la idea de que SVM produce un hiperplano en particular en el espacio de características que separa los compuestos en activos e inactivos, en el llamado el margen máximo del hiperplano (Figura 1.8) . Los Kernels más utilizados dentro de SVM son: lineal (punto) , polinómica, Neural (sigmoide, Tanh), Anova, Fourier, Spline ,B Spline, Aditivo, Tensor y Gaussian Radial Basis o de forma Exponencial Radial.

En un corto período de tiempo, se han descubierto numerosas aplicaciones de los SVM tanto en la química, como en el diseño de fármacos (que discriminan entre ligandos y no ligandos , inhibidores y no inhibidores, etc.) [137], y en el descubrimiento de fármacos [138] .

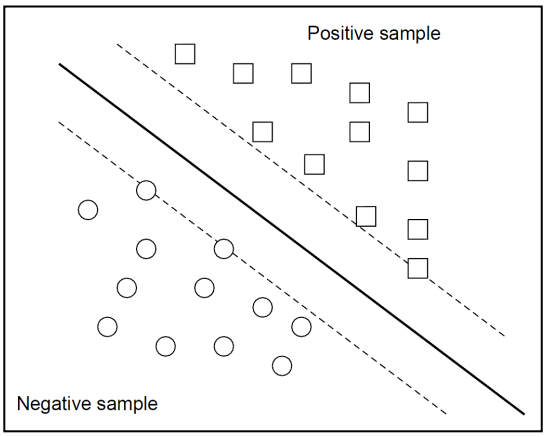


Figura 1.8. Márgenes de los Hiperplanos en las Maquina de Soporte Vectorial

1.4.4.3 BOSQUES ALEATORIOS

El método RF está siendo utilizado de una manera extensiva en multitud de campos de investigación, tanto para seleccionar aquellas variables con mayor poder clasificador de entre un conjunto, como para clasificar conjuntos de datos. En RF cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. En RF cada árbol individual se explora de una manera particular:

1. Dado un conjunto de datos de entrenamiento *N*, se toman *N* muestras aleatorias con repetición (*Bootstrap*) como conjunto entrenamiento.
2. Para cada nodo del árbol, se determinan *M* variables de entrada, y se determina "*m*" << *M*, para cada nodo, seleccionando *m* variables aleatorias. La variable mas relevante elegida al azar se

usa en el nodo. El valor de *m* se mantiene constante durante la expansión del bosque.

3. Cada árbol es desarrollado hasta su expansión máxima, nunca se poda.

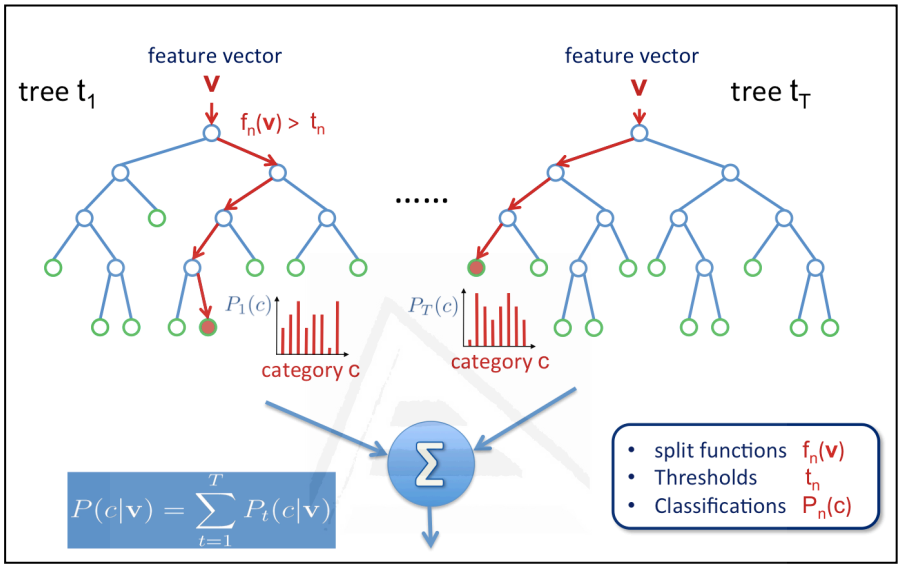


Figura 1.9. Espacio de soluciones de Random Forest. A partir de un vector de características, se construyen un conjunto de árboles.

El error del conjunto de árboles depende de dos factores.

- Correlación entre dos árboles cualesquiera del bosque, que se evita con la utilización de un subconjunto de variables elegidas al azar y el remuestreo de datos (*Boostrap*)
- Clasificador Fuerte, la importancia de cada árbol del bosque, se denota con un error bajo de este, el incremento de estos clasificadores decrementa el error del bosque.

1.4.5 SELECCIÓN AUTOMÁTICA DE CARACTERÍSTICAS

Para que el sistema pueda tomar una decisión sobre la clasificación, es necesario indicar las características de entrada, las variables predictores y el mecanismo de selección de las características usadas para la clasificación [139]. El conjunto de características que describe un objeto puede ser arbitrariamente grande, por lo que en la mayoría de las ocasiones se hace necesaria una etapa previa de selección de características.

Los conjuntos de datos de entrada que corresponden a un dataset representan un numero fijo de características, en nuestro ámbito de dominio estos son los descriptores moleculares, los valores de estos predictores pueden ser binarios, categóricos o continuos y forman parte del conjunto de datos entrada del sistema.

El proceso de selección de características consta de dos pasos principales: la constitución de los datos (filtrado, idoneidad, escalado) y la selección de características. ¿Cuáles son características mas relevantes para nuestro dominio de aplicación?, como estamos tratando bases de datos normalizadas, obviamos el filtrado, la idoneidad y el escalado de estos datos. Nos centraremos en la selección de características. Existen diferentes motivaciones para hacerlo, pero buscamos obtener una serie de ventajas [140], dependiendo del método empleado, buscando obtener algunos de los beneficios siguientes:

- *Reducción del numero de datos totales*, así como la reducción y la dimensión de la información global.
- *Reducción de características*, disminuye el coste de almacenamiento continuo.

- *Mejora del rendimiento*, la mejora de la velocidad puede conllevar una mejora en la precisión de la predicción.
- *Mejoras en la visualización*, el entendimiento de la información disponible para el problema.
- *Reducción del tiempo de entrenamiento de los modelos*, subconjunto más reducido en número de datos.
- *Reducción del ruido en los datos*, eliminando las características irrelevantes o redundantes.

1.4.5.1 SELECCIÓN AUTOMÁTICA DE DESCRIPTORES

La selección adecuada del conjunto de los descriptores moleculares (predictores) es fundamental para optimizar la predicción y la selección automática de estos descriptores es un claro objetivo frente a una selección manual (*ad hoc*). ¿Qué variables son las más importantes en los modelos de clasificación?, esta problemática es la habitual en muchos dominios de investigación. Esto se solventa habitualmente empleando las variables que mejor pueden explicar nuestro modelo y se adapten al dominio en el que estamos. Para algunos dominios, se hacen segmentaciones en base al sexo y la edad o se construyen variables artificiales (*dummy*). Estos son los mecanismos propios y adoptados por un gran conocedor del dominio en el que nos encontramos y en ocasiones es una tarea multidisciplinar, y en todo caso construida para cada problema particular (ad hoc), que tratemos de predecir o clasificar. La utilización de técnicas de inteligencia computacional, nos permitirá seleccionar estas variables de una forma automática cuantificando la importancia relativa de las variables. RF es un método de clasificación basado en la realización de múltiples árboles de decisión sobre muestras

de un conjunto de datos. La posibilidad de incluir un gran número de variables de entrada en nuestro modelo (predictores) permite encontrar relaciones lineales entre ellas y evitar las que aparecerán debidas al azar, esto hace al método muy interesante para este fin.

1.4.5.2 SUBCONJUNTO MÍNIMO DE CARACTERÍSTICAS

Una vez introducida la idea de la relevancia de las características seleccionadas, aquellas no seleccionadas, o que han quedado fuera deben ser las irrelevantes o redundantes. Por tanto, el orden de relevancia nos permite extraer un subconjunto mínimo de características que son suficientes para hacer una predicción óptima.

Con la selección de variables ya realizada, podemos entrenar de nuevo el modelo con las bases de datos de compuestos conocidos activos o inactivos. Esta información será utilizada posteriormente para mejorar las predicciones y contribuir a que otros métodos puedan posteriormente acelerar el descubrimiento de nuevos fármacos aplicando las técnicas de CV.

Frente a la selección manual de descriptores moleculares que utilizan sólo un pequeño conjunto de propiedades químicas representativas, la selección automática emplea un conjunto mayor de estas por su campo de exploración y la comparación es mucho mayor, permitiendo un mejor ajuste y bondad en la predicción de la actividad.

PUBLICACIONES DERIVADAS

Esta tesis presentada por compendio representa los resultados de la investigación realizada en los últimos 3 años que dieron como fruto las siguientes publicaciones en revistas internacionales impactadas en el Journal Citation Report:

- **Improvement of Virtual Screening predictions using Computational Intelligence methods**. Gaspar Cano, José García-Rodríguez and Horacio Pérez-Sánchez. Letters in Drug Design & Discovery, 11, 33-39, 2014. JCR IMPACT FACTOR: 0,845.

- **Improving Drug Discovery using Hybrid Softcomputing Methods.** Horacio Pérez-Sánchez, Gaspar Cano and José García-Rodríguez. Applied SoftComputing Journal, 2013, JCR IMPACT FACTOR: 2,140. http://dx.doi.org/doi:10.1016/j.asoc.2013.10.033.

2.1 IMPROVEMENT OF VIRTUAL SCREENING PREDICTIONS USING COMPUTATIONAL INTELLIGENCE METHODS

Letters in Drug Design & Discovery, 2014, 11, 33-39

Resumen: Los métodos de Cribado Virtual (CV) pueden ayudar considerablemente en la investigación clínica, prediciendo cómo interactúan los ligandos candidatos a fármacos con sus dianas farmacológicas. Sin embargo, la exactitud de la mayoría de los métodos de Cribado Virtual se encuentra sujeta a las limitaciones de las funciones de "*scoring*" (que ponderan las interacciones biomoleculares) e incertidumbres que hoy por hoy no se conoce completamente. Con el fin de mejorar la precisión de estas funciones de ponderación, utilizadas en la mayoría de los métodos de CV, se propone un nuevo enfoque híbrido mediante el uso de las Redes Neuronales (*NNET*) y Máquinas de Soporte Vectorial (*SVM*), entrenadas con bases de datos de principios activos y no inactivos (fármacos). De forma que esta información será utilizada posteriormente para mejorar las predicciones del Cribado Virtual.

Send Orders for Reprints to reprints@benthamscience.net

Letters in Drug Design & Discovery, **2014,** *11,* **33-39**

Improvement of Virtual Screening Predictions using Computational Intelligence Methods

Gaspar Cano[1], José García-Rodríguez[1] and Horacio Pérez-Sánchez[2,*]

[1]*Computing Technology Department, University of Alicante, Ap. 99. E03080. Alicante, Spain*

[2]*Computer Science Department, Catholic University of Murcia (UCAM) E30107 Murcia, Spain*

Abstract: Virtual Screening (VS) methods can considerably aid clinical research, predicting how ligands interact with drug targets. However, the accuracy of most VS methods is constrained by limitations in the scoring function that describes biomolecular interactions, and even nowadays these uncertainties are not completely understood. In order to improve accuracy of scoring functions used in most VS methods we propose a hybrid novel approach where neural networks (NNET) and support vector machines (SVM) methods are trained with databases of known active (drugs) and inactive compounds, this information being exploited afterwards to improve VS predictions.

Keywords: Clinical Research, Computational Intelligence, Drug Discovery, Neural Networks, Support Vector Machines, Virtual Screening.

1. INTRODUCTION

In clinical research, it is crucial to determine the safety and effectiveness of current drugs and to accelerate findings in basic research (discovery of new leads and active compounds) into meaningful health outcomes. Both objectives need to process the large data set of protein structures available in biological databases such as PDB [1] and also derived from genomic data using techniques such as homology modeling [2]. Screenings in lab and compound optimization are expensive and slow methods, but bioinformatics can vastly help clinical research for the mentioned purposes by providing prediction of the toxicity of drugs and activity in non-tested targets, and by evolving discovered active compounds into drugs during clinical trials.

This aim can be achieved thanks to the availability of bioinformatics tools and Virtual Screening (VS) methods that allow testing all required hypothesis before clinical trials. However, the accuracy of most VS methods is constrained by limitations in the scoring function that describes biomolecular interactions, and even nowadays these uncertainties are not completely understood. In order to solve this problem we propose a novel hybrid approach where Computational Intelligence (CI) methods that include neural networks (NNET) and support vector machines (SVM) are trained with databases of known active (drugs) and inactive compounds (decoys) and later used to improve VS predictions. Other approaches based on the use of molecular descriptors have been previously described in the literature but they were applied in concrete contexts of protein-ligand interactions [2-4], while the method we propose can be applied to any case of protein-ligand interactions and VS method, provided previous experimental information for active and inactive compounds is available.

*Address correspondence to this author at the Computer Science Department, Catholic University of Murcia (UCAM) E30107 Murcia, Spain;
Tel: 0034-968277982; Fax: 0034-968277943; E-mail: hperez@ucam.edu

The rest of the paper is organized as follows. Section 2 describes the methodology including VS, NNET and SVM techniques, and molecular properties used in this study. Section 3 presents the experiments carried out to refine the VS methods with the previously mentioned techniques while section 4 reports the results obtained. In section 5 we present our main conclusions and further work.

2. METHOLODOGY

In this section we describe the methodologies we used for the improved prediction of protein-ligand affinities; a) the Virtual Screening method (VS), and b) two different CI techniques are employed that include; neural networks (NN) and support vector machines (SVM) trained with different molecular properties calculated for known active and inactive compounds selected from standard VS benchmarks. In Fig. (**1**) the flowchart of our experimental setup is depicted.

2.1. Virtual Screening

Essentially, VS methods screen a large database of molecules in order to find compounds that fit some established criteria [6]. In the case of the discovery of new leads, compound optimization, toxicity evaluation and additional stages of the drug discovery process, we screen a large compound database to find a small molecule which interacts in a desired way with one or many different receptors. Among the many available VS methods for this purpose one of the most structurally accurate methods is protein-ligand docking [7, 8]. These methods try to obtain rapid and accurate predictions of the 3D conformation a ligand adopts when it interacts with a given protein target, and also the strength of this union, in terms of its scoring function value. Docking simulations are typically carried out in a very concrete part of the protein surface in methods such as Autodock [9], Glide [10] and DOCK [11], to name a few. This region is commonly derived from the position of a particular ligand in the crystal structure, or from the crystal structure of the protein without any ligand. The former can be performed when the protein is

1875-628X/14 $58.00+.00

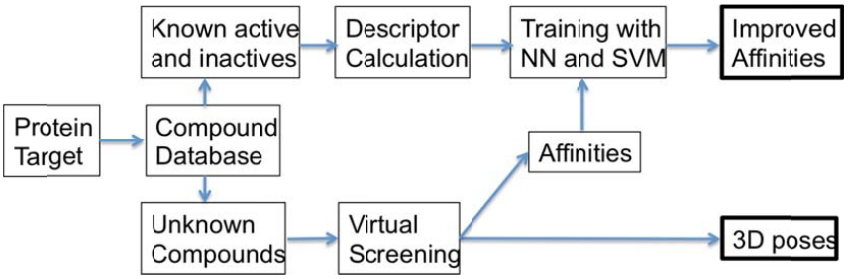


Fig. (1). Flowchart of the experimental setup used in this work.

co-crystallized with the ligand, but it might happen that no crystal structure of this ligand-protein pair is at disposal. Nevertheless, the main problem is to take the assumption, once the binding site is specified, that many different ligands will interact with the protein in the same region, discarding completely the other areas of the protein.

In essence, in a docking simulation we calculate the ligand-protein interaction energy for a given starting configuration of the system, which is represented by a scoring function [12]. In most VS methods the scoring function calculates electrostatic (ES), Van der Waals (VDW) and hydrogen bond (HBOND) terms.

Furthermore, in docking methods it is normally assumed [13] that the minima of the scoring function, among all ligand-protein conformations, will accurately represent the conformation the system adopts when the ligand binds to the protein. Thus, when the simulation starts, we try to minimize the value of the scoring function by continuously performing random or predefined perturbations of the system, calculating for each step the new value of the scoring function, and accepting it or not following different approaches like the Monte Carlo minimization method [14, 15].

2.2. Computational Intelligence Methods

We describe in this section the CI methods we will apply to refine the prediction capacities of VS.

2.2.1. Neural Networks

One of the most dominant application areas of neural networks is non-linear function approximation. The main advantage of neural network modeling is that complex non-linear relationships can be modeled without assumptions about the form of the model. That feature is very useful in the field of drug design and drug discovery.

More than two decades ago, the aqueous solubility of organic compounds was studied using neural approaches [16]. In next decade, supervised and unsupervised neural models were employed to model QSAR, predict molecules activities and structure, clustering and many more [17, 18]. More recently the problem of drug solubility prediction from structure has been revisited [19]. Properties of organic compounds obtained from the molecular structure have been extensively studied using hybrid techniques that include neural networks [20-22]. Also identification of small-molecule ligands has been improved using neural techniques [23-25]. In the last years a large number of authors have designed hybrid methods that combined neural networks with other techniques to solve chemistry related problems.

There are several types of feed-forward neural networks (NNET), the most widely used being multi-layer networks with sigmoidal activation functions (multi-layer perceptrons) and single layer networks with local activation functions (radial basis function networks). The good approximation capability of neural networks has been widely demonstrated by both practical applications and theoretical research. We decided to use a single-hidden-layer neural network with skip-layer connections in this study, as shown in Fig. (**2**), since it has been clearly demonstrated its impact on the differentiation between active and inactive compounds and other chemical applications [17]. For such purpose we used the *nnet* function of the R package [26].

2.2.2. Support Vector Machines

Support vector machines (SVM) [27] are a group of supervised learning methods that can be applied to classification or regression. They represent the decision boundary in terms of a typically small subset of all training examples, called the support vectors. In a short period of time, SVM have found numerous applications in chemistry, such as in drug design [28] when discriminating between ligands and non-ligands, inhibitors and non-inhibitors, drug discovery [29], quantitative structure-activity relationships (QSAR, where SVM regression is used to predict various physical,

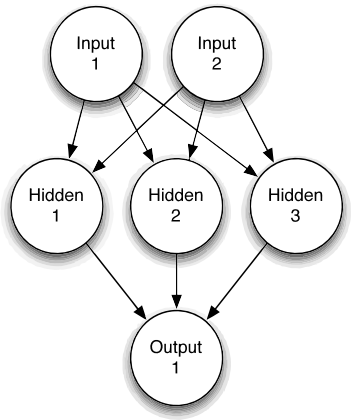


Fig. (2). Single Hidden layer Neural Network.

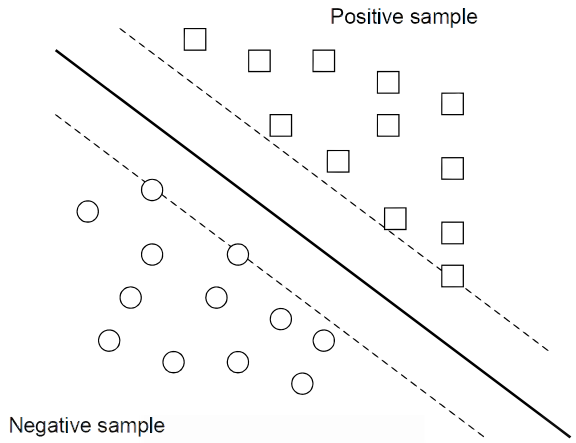


Fig. (3). Support Vector Machines margin hyperplanes.

Table 1. Number of Active (Ligands) and Inactive Compounds (Decoys) for each of the Sets used in this Study, Obtained from DUD [33].

Protein	PDB Code	Resolution (Å)	n_o of Ligands	n_o of Decoys
GPB	1A8I	1.8	52	1851
MR	2AA2	1.9	15	535
TK	1KIM	2.1	22	785

chemical, or biological properties) [30], chemometrics (optimization of chromatographic separation or compound concentration prediction from spectral data as examples), sensors (for qualitative and quantitative prediction from sensor data), chemical engineering (fault detection and modeling of industrial processes)[31]. An excellent review of SVM applications in chemistry can be found in [32].

In our case, we exploit the idea that SVM produce a particular hyperplane in feature space that separates the active from the inactive compounds called the maximum margin hyperplane, as shown in Fig. (**3**).

Most used kernels within SVM technique include: linear (dot), Polynomial, Neural (sigmoid,Tanh), Anova, Fourier, Spline, B Spline, Additive, Tensor and Gaussian Radial Basis or Exponential Radial Basis.

2.3. Ligand Databases and Molecular Properties

We carried out our study applying the methods described in sections 2.2.1 and 2.2.2 and using different sets of molecules that are known to be active or inactive. We employed standard VS benchmark tests, such as the Directory of Useful Decoys (DUD) [33], where VS methods check how efficient they are in differentiating ligands that are known to bind to a given target, from non-binders or decoys. Input data for each molecule of each set contains its molecular structure and whether it is active or not. We focused on three diverse DUD datasets (details are shown in Table **1**) that cover kinases, nuclear hormone receptors and other enzymes such as TK, which corresponds to thymidine kinase (from PDB 1KIM), MR, which corresponds to mineralocorticoid receptor (from PDB 2AA2), and GPB, which corresponds to the enzyme glycogen phosphorylase (from PDB 1A8I).

Next, using the ChemoPy package [34] we calculated for all ligands of the TK, MR and GPB sets a diverse set of molecular properties derived from the set of constitutional, CPSA (charged partial surface area) and fragment/fingerprint-based descriptors, as described in Table **2**. Constitutional properties depend on very simple descriptors of the molecule that can be easily calculated just counting the number of molecular elements such as atoms, types of atoms, bonds, rings, etc. These descriptors should be able to differentiate very dissimilar molecules, but might have problem for separating closely related isomers. CPSA descriptors take into account finer details of molecular structure, so they might be able to separate similar molecules, but might also have difficulties for separating isomers. Lastly, fragment and fingerprint-based descriptors take into account the presence of an exact structure (not a substructure) with limited specified attachment points. These descriptors are more difficult to calculate. In generating the fingerprints, the program assigns an initial code to each atom. The initial atom code is derived from the number of connections to the atom, the element type, atomic charge, and atomic mass. This corresponds to an ECFP with a neighborhood size of zero. These atom codes are then updated in an iterative manner to reflect the codes of each atoms neighbors. In the next iteration, a

Table 2. Molecular Descriptors used in this Study.

Constitutional Descriptors	
Natom	Number of atoms
MolWe	Molecular Weight
NRing	Number of rings
NArRg	Numer of aromatic rings
NRotB	Number of rotatable bonds
NHDon	Number of H-bond donors
NHAcc	Number of H-bond acceptors
Cpsa Descriptors	
Msurf	Molecular surface area
Mpola	Molecular polar surface area
Msolu	Molecular solubility
AlogP	Partition coefficient
Fragment/fingerprint-Based descriptors	
ECP2, ECP4, ECP6	Extended-connectivity fingerprints (ECFP)
EstCt	Estate counts
AlCnt	AlogP2 Estate counts
EstKy	Estate keys
MDLPK	MDL public keys

hashing scheme is employed to incorporate information from each atoms immediate neighbors. Each atoms new code now describes a molecular structure with a neighborhood size of one. This process is carried out for all atoms in the molecule. When the desired neighborhood size is reached, the process is complete and the set of all features is returned as the fingerprint. For the ECFPs employed in this paper, neighborhood sizes of two, four and six (ECFP 2, ECFP 4, ECFP 6) were used to generate the fingerprints. The resulting ECFPs can represent a much larger set of features than other fingerprints and contain a significant number of different structural units crucial for the molecular comparison, among the compounds.

3. RESULTS

A set of experiments has been carried out in order to test the validity of our initial hypothesis combining and refining VS results with the proposed CI methods.

3.1. Activity Prediction Using Computational Intelligence Methods

NNET and SVM were trained with the previously described DUD datasets TK, MR and GPB. Molecular properties described in Table **2** were calculated for each molecule as described in the methods section.

A *k*-fold cross-validation technique with *k*=5 was employed for NNET and SVM experiments.

3.1.1. NNET

A set of experiments has been developed to find the feed-forward neural network architecture that fits better to the problem of classification proposed. A combination of different number of neurons for the hidden layer has been tested with the different descriptors and datasets. We considered architectures with 1, 2 and 3neurons in the hidden layer. Since results of combinations with more than 3 neurons did not improve the results, we decide to use the simplest option with 3 neurons due to its lower temporal cost for training phase. Results for AUC values are reported in Fig (**4**).

3.1.2. SVM

A set of experiments with different kernels has been developed to find the option with higher discrimination capacities between active and non-active compounds for each descriptor. More specifically, linear, polynomial, sigmoid and radial kernels has been tested with all the descriptors and datasets. Best results have been obtained with radial kernels and results obtained for AUC values are reported in Fig. (**5**).

4. DISCUSSION

AUC values reported by both NNET and SVM depend clearly on the considered molecular property, and to a lesser extent, on the molecular dataset studied (GPB, MR, TK). The reason for the latter might be that main active compounds of these sets have similar structures, as shown in Fig. (**6**), consisting in small molecules with two or four rings, and

Table 3. Combinations of Molecular Descriptors Used in this Study.

Combinations Of Constitutional Descriptors	
MNBH	Molecular polar surface area (MPola)+ Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MNB	Molecular polar surface area (MPola) + Number of rotatable bonds (NRotB)
NBH	Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
Combinations Of Fragment/Fingerprint-Based Descriptors	
EAE246	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EA	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt)
AE246	AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EE246	Estate counts (EstCt) + Extended-connectivity fingerprints (ECFP)

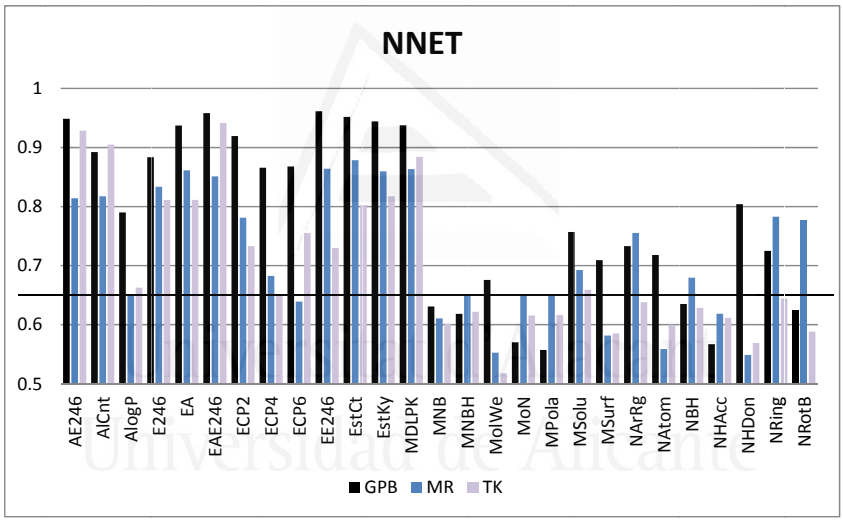


Fig. (4). AUC values of the ROC curves obtained using NNET as described in section 2.2.1 for each property of Table **3** of the three different datasets GPB (black), MR (blue) and TK (light grey). Baseline for AUC=0.65 is also shown. The resulting AUC values for the combined properties described in Table **4** are also reported.

also because they establish similar interactions with the protein, mainly based on hydrogen bond networks.

We propose a threshold value of 0.65 for AUC in order to discriminate which properties are useful for active/inactive prediction. Properties that simultaneously yield AUC values higher than this threshold for all sets using both NNET and SVM are; AlCnt, E246, ECP2 and MDLPK, while properties that yield AUC values lower than threshold are mostly AlogP, MolWe, MPola, MSolu, MSurf, NArgRg, Natom, NHacc, NHDon, NRing, and NRotB. So it seems clear that the best option for discriminating among active and inactive compounds in these datasets is to use fingerprint-based descriptors and to avoid the use of constitutional and CPSA descriptors. This is reasonable since fingerprint descriptors take into account more details about the structure of molecules, being able to efficiently discriminate with more accuracy between active compounds and their decoys.

Next, we studied whether combination of properties could lead to improvements on the predictive capability of these CI methods. Therefore we combined properties that yielded the lowest AUC values, constitutional descriptors, and the properties that yielded the highest AUC values, so fingerprint based descriptors. Combinations used are described in Table **3** and AUC values obtained are reported in Figs. (**4**) and (**5**). In the case of combinations of constitutional descriptors, there is no clear improvement for either NNET or SVM, while for fingerprint combinations, average AUC values for the three datasets improve slightly.

Finally, top obtained AUC values for datasets GPB, MR and TK correspond to properties EE246 (0.96), EstCt (0.87)

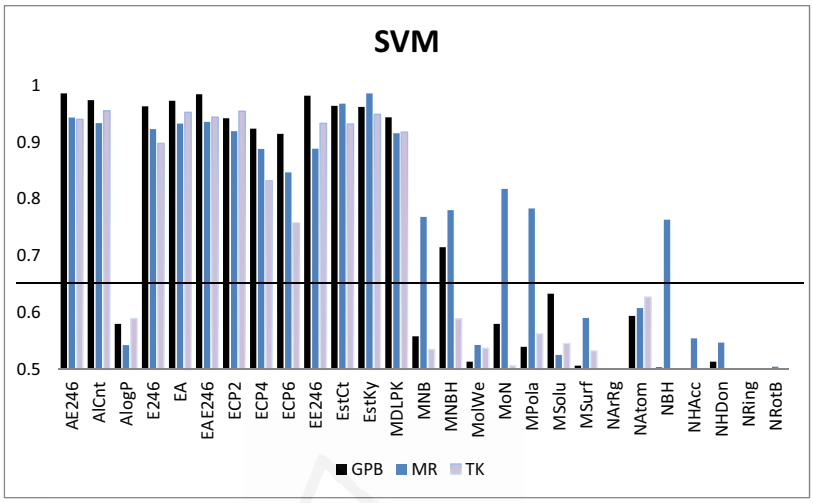


Fig. (5). AUC values of the ROC curves obtained using SVM as described in section 2.2.1 for each property of Table **2** of the three different datasets GPB (black), MR (blue) and TK (light grey). Baseline for AUC=0.65 is also shown. The resulting AUC values for the combined properties described in Table **3** are also reported.

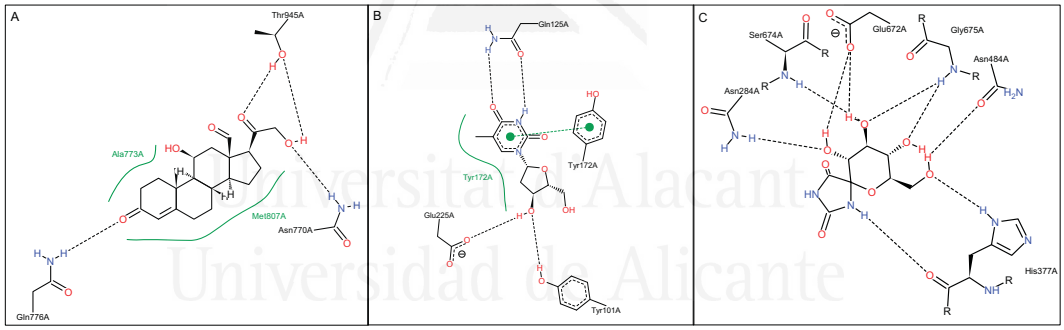


Fig. (6). Depiction of the molecular structure and protein-ligand interactions established by main active compounds from **A**) MR, **B**) TK, and **C**) GPB.

and EAE246 (0.94) when using NNET, and AE246 (0.98), EstKy (0.98)and AlCnt (0.95) when using SVM.

Consequently, and taking into account information obtained by CI methods we can post-process docking results obtained by the scoring function of VS methods and neglect resulting compounds that are predicted as inactive. Then we can sort them by the final affinity value predicted by the VS scoring function for such cases and study visually the top ones.

5. CONCLUSION

In this work we have shown how the predictive capability of the VS methods can be improved using CI methods such as neural networks and support vector machines. It must be mentioned that CI approaches can only be used when experimental data for active and non-active compounds for a given protein is available.

This methodology can be used to improve drug discovery, drug design, repurposing and therefore aid considerably in clinical research. In the next steps we want to extend our ideas to the application of unsupervised CI methods.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

We thank the Catholic University of Murcia (UCAM) under grant PMAFI/26/12. This work was partially supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain. The authors also thankfully acknowledge

the computer resources and the technical support provided by the Plataforma Andaluza de Bioinformática of the University of Málaga.

REFERENCES

[1] Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nuc. Acids Res.,* **2000**, *28*, 235–242.

[2] Agatanovic-Kustrin, S.; Turner, J.V. Artificial neural network modeling of phytoestrogen binding to estrogen receptors. *Lett. Drug Des. Discover.,* **2006**, *7*, 436–442.

[3] Khadikar, P.V.; Deeb, O.; Jaber, A.; Singh, J.; Agrawal, V.K.; Singh, S.; Lakhwani, M. Development of quantitative structure-activity relationship for a set of carbonic anhydrase inhibitors: Use of quantum and chemical descriptors. *Lett. Drug Des. Discover.,* **2006**, *3*, 622–635.

[4] Mishra, N. K.; Raghava, G.P.S. Prediction of specificity and cross reactivity of kinase inhibitors. *Lett. Drug Des. Discover.,* **2011**, *8*, 223–228.

[5] Sanchez, R.; Sali, A. Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. *PNAS.,* **1998**, *95*, 13597–13602.

[6] Jorgensen, W. L. The many roles of computation in drug discovery. *Science.,* **2004**, *303*, 1813–8.

[7] Yuriev, E.; Agostino, M.; Ramsland, P. A. Challenges and advances in computational docking: 2009 in review. *J. Mol. Recognit.,* **2011**, *24*, 149–164.

[8] Huang, S.-Y.; Zou, X. Advances and challenges in protein-ligand docking. *Int. J. Mol. Sci.,* **2010**, *11*, 3016–34.

[9] Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comp. Chem.,* **1998** *19*, 1639–1662.

[10] Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.,* **2004** *47*, 1739–1749.

[11] Ewing, T. J.; Makino, S.; Skillman, a G.; Kuntz, I. D. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *J. Comput. Aided Mol. Des.,* **2001**, *15*, 411–28.

[12] Wang, R.; Lu, Y.; Fang, X.; Wang, S. An extensive test of 14 scoring functions using the PDBbind refined set of 800 protein-ligand complexes. *J. Chem. Inf. Comput. Sci.,* **2004,** *44*, 2114–2125.

[13] Jorgensen, W. L. The many roles of computation in drug discovery. *Science.,* **2004**, *303*, 1813–1818.

[14] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.,* **1953**, *21*, 1087–1092.

[15] Sánchez-Linares, I.; Pérez-Sánchez, H.; Cecilia, J. M.; García, J. M. High-Throughput parallel blind Virtual Screening using BINDSURF. *BMC Bioinformatics.,* **2012**, *13*, S13.

[16] Bodor, N.; Harget, A.; Huang, M.J. Neural network studies. 1. Estimation of the aqueous solubility of organic compounds. *J. Am. Chem. Soc.*, **1991**, *113*, 9480–9483.

[17] Schneider, G.; Wrede, P. Artificial neural networks for computer-based molecular design. *Prog. Biophys. Mol. Biol.,* **1998**, *70*, 175–222.

[18] Peterson, K.L. Artificial Neural Networks and Their use in Chemistry. *Rev. Comp. Ch.,* **2007**, *16*, 53-140.

[19] Jorgensen, W.L; Duffy, E.M. Prediction of drug solubility from structure. *Adv. Drug. Deliv. Rev.,* **2002**, *54*, 355-66.

[20] Taskinen, J; Yliruusi, J. Prediction of physicochemical properties based on neural network modelling. *Adv. Drug Deliv. Rev.,* **2003**, *55*, 1163–1183.

[21] Durrant, J. D.; McCammon, J. A. NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein-Ligand Complexes. *J. Chem. Inf. Model.*, **2010**, *50*, 1865–1871.

[22] Durrant, J. D.; McCammon, J. A. NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function. *J. Chem. Inf. Model.,* **2011**, *51*, 2897–2903.

[23] Weisel, M.; Kriegl, J.M.; Schneider, G. Architectural repertoire of ligand-binding pockets on protein surfaces. *Chembiochem*., **2010**, *11*, 556-563.

[24] Pal, N.R.; Panja, R. Finding short structural motifs for re-construction of proteins 3D structure. *Appl. Soft Comput*., **2013**, *13*, 1214–1221.

[25] Romero Reyes, I.V.; Fedyushkina, I.V.; Skvortsov, V.S.; Filimonov, D.A Prediction of progesterone receptor inhibition by high-performance neural network algorithm. Internat. *J. Math. Model. Methods. Appl. Sci*., **2013**, *7*, 303-310.

[26] Venables, W. N.; Ripley, B. D. *MASS: Modern Applied Statistics with S,* 4th ed.; Springer: New York, **2002**

[27] Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.,* **1995**, *20*, 273–297.

[28] Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases Using a Support Vector Machine. *J. Chem. Inf. Model.,* **2005**, *45*, 549–561.

[29] Warmuth, M.K.; Liao, J; Rätsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci*., **2003**, *43*, 667-73.

[30] Kriegl, J. M.; Arnhold, T.; Beck, B.; Fox, T. Prediction of Human Cytochrome P450 Inhibition Using Support Vector Machines. *QSAR Comb. Sci.,* **2005**, *24*, 491–502.

[31] Lee, D. E.; Song, J.-H.; Song, S.-O.; Yoon, E. S. Weighted Support Vector Machine for Quality Estimation in the Polymerization Process. *Ind. Eng. Chem. Res.,* **2005**, *44*, 2101–2105.

[32] Ivanciuc, O. Applications of Support Vector Machines in Chemistry. *Rev. Comp. Ch.,* **2007**, *2*. 291–400.

[33] Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.,* **2006**, *49*, 6789–6801.

[34] Cao, D.-S.; Xu, Q.-S.; Hu, Q.-N.; Liang, Y.-Z. ChemoPy: freely available python package for computational biology and chemoinformatics. *Bioinformatics.,* **2013**, *29*, 1092–1094.

Received: May 19, 2013 Revised: July 12, 2013 Accepted: July 30, 2013

2.2 IMPROVING DRUG DISCOVERY USING HYBRID SOFTCOMPUTING METHODS

Applied Soft Computing 20 (2014) 119–126

Resumen: Los métodos de Cribado Virtual (CV) pueden ayudar considerablemente la investigación clínica, prediciendo cómo interactúan los ligandos con su objetivos farmacológicos. La mayoría de los métodos de CV suponen un único punto de unión del ligando con la proteína diana, pero se ha demostrado que diversos ligandos pueden interactuar con partes no relevantes del objetivo y que muchos métodos de CV no lo tienen en cuenta, siendo este un hecho relevante. Este problema se evita mediante una nueva metodología, BINDSURF que explora toda la superficie de la proteína con el fin de encontrar nuevos puntos de unión, donde los ligandos pueden potencialmente interactuar con la proteína de interés, para esto se utiliza la última generación de hardware masivamente paralelo GPU, esto permite un rápido procesamiento de grandes bases de datos de ligandos. BINDSURF, por lo tanto, se puede utilizar en el descubrimiento, diseño y reutilización de fármacos, y ayudar considerablemente en la investigación clínica. Sin embargo, la exactitud de la mayoría de los métodos de CV y concretamente BINDSURF está restringido por limitaciones en función de la ponderación que describe las interacciones biomoleculares, e incluso hoy en día estas incertidumbres no se conoce completamente. Con el fin de mejorar la precisión de las función de ponderación utilizados en BINDSURF, proponemos un nuevo enfoque híbrido en el que las redes neuronales (NNET) y las máquinas de soporte vectorial (SVM), están entrenados con bases de datos conocidas de compuestos (fármacos) activos e inactivos, permitiendo que esta información sea utilizada posteriormente para mejorar las predicciones de CV BINDSURF.

Applied Soft Computing 20 (2014) 119–126

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Improving drug discovery using hybrid softcomputing methods

Horacio Pérez-Sánchez [a,*], Gaspar Cano [b], José García-Rodríguez [b]

[a] Computer Science Department, Catholic University of Murcia (UCAM), E30107 Murcia, Spain
[b] Computing Technology Department, University of Alicante, Ap. 99, E03080 Alicante, Spain

ARTICLE INFO

Article history:
Received 29 March 2013
Received in revised form 29 October 2013
Accepted 30 October 2013
Available online 28 November 2013

Keywords:
Neural networks
Support vector machines
Clinical research
Drug discovery
Virtual screening
Parallel computing

ABSTRACT

Virtual screening (VS) methods can considerably aid clinical research, predicting how ligands interact with drug targets. Most VS methods suppose a unique binding site for the target, but it has been demonstrated that diverse ligands interact with unrelated parts of the target and many VS methods do not take into account this relevant fact. This problem is circumvented by a novel VS methodology named BINDSURF that scans the whole protein surface in order to find new hotspots, where ligands might potentially interact with, and which is implemented in last generation massively parallel GPU hardware, allowing fast processing of large ligand databases. BINDSURF can thus be used in drug discovery, drug design, drug repurposing and therefore helps considerably in clinical research. However, the accuracy of most VS methods and concretely BINDSURF is constrained by limitations in the scoring function that describes biomolecular interactions, and even nowadays these uncertainties are not completely understood. In order to improve accuracy of the scoring functions used in BINDSURF we propose a hybrid novel approach where neural networks (NNET) and support vector machines (SVM) methods are trained with databases of known active (drugs) and inactive compounds, being this information exploited afterwards to improve BINDSURF VS predictions.

1. Introduction

In clinical research, it is crucial to determine the safety and effectiveness of current drugs and to accelerate findings in basic research (discovery of new leads and active compounds) into meaningful health outcomes. Both objectives need to process the large data set of protein structures available in biological databases such as PDB [1] and also derived from genomic data using techniques as homology modeling [2]. Screenings in lab and compound optimization are expensive and slow methods [3], but bioinformatics can vastly help clinical research for the mentioned purposes by providing prediction of the toxicity of drugs and activity in non-tested targets, and by evolving discovered active compounds into drugs for the clinical trials.

This can be achieved thanks to the availability of bioinformatics tools and Virtual Screening (VS) methods that allow testing all required hypothesis before clinical trials. Nevertheless current Virtual Screening (VS) methods, such as docking, fail to make good toxicity and activity predictions since they are constrained by the access to computational resources; even the nowadays fastest VS methods cannot process large biological databases in a reasonable time-frame. Therefore, these constraints impose serious limitations in many areas of translational research.

The use of last generation massively parallel hardware architectures such as Graphics Processing Units (GPUs) can tremendously overcome this problem. The GPU has become increasingly popular in the high performance computing arena, by combining impressive computational power with the demanding requirements of real-time graphics and the lucrative mass-market of the gaming industry [4]. Scientists have exploited this power in arguably every computational domain, and the GPU has emerged as a key resource in applications where parallelism is the common denominator [5]. To maintain this momentum, new hardware features have been progressively added by NVIDIA to their range of GPUs, with the Fermi architecture [6] being the most recent milestone in this path. Therefore, GPUs are well suited to overcome the lack of computational resources in VS methods, accelerating the required calculations and allowing the introduction of improvements in the biophysical models not a affordable in the past [7]. We have previously worked in this direction, showing how VS methods can benefit from the use of GPUs [8,9,10]. Moreover, another important lack of VS methods is that they usually take the assumption that the binding site derived from a single crystal structure will be the same for different ligands, while it has been shown that this does not always happen [11], and thus it is crucial to avoid this very basic supposition. In this work, we present a novel VS methodology called BINDSURF [12] which takes advantage of massively parallel

* Corresponding author. Tel.: +34 653067844.
E-mail addresses: hperez@ucam.edu (H. Pérez-Sánchez), gcano@dtic.ua.es (G. Cano), jgarcia@dtic.ua.es (J. García-Rodríguez).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.10.033

and high arithmetic intensity of GPUs to speed-up the required calculations in low cost and consumption desktop machines, providing new and useful information about targets and thus improving key toxicity and activity predictions. In BINDSURF a large ligand database is screened against the target protein over its whole surface simultaneously. Afterwards, information obtained about novel potential protein hotspots is used to perform more detailed calculations using particular VS method, but just for a reduced and selected set of ligands.

Other authors have also performed VS studies over whole protein surfaces [13] using different approaches and screening small ligand databases, but as far as we know, none of them have been implemented on GPUs, while BINDSURF has been designed from scratch taken into account the GPU architecture.

However, the accuracy of most VS methods is constrained by limitations in the scoring function that describes biomolecular interactions, and even nowadays these uncertainties are not completely understood. In order to solve this problem we propose a novel hybrid approach where softcomputing methods that includes neural networks (NNET) and support vector machines (SVM) are trained with known active (drugs) and inactive compounds and are later used to improve VS predictions.

The rest of the paper is organized as follows. Section 2 describes the methodology including VS using BINDSURF, NNET and SVM techniques, and molecular properties used in this study. Section 3 presents the experiments carried out to refine the BINDSURF method with the previously mentioned techniques while Section 4 discusses the results obtained. In Section 5 we present our main conclusions and further work.

2. Metholodogy

In this section we describe the methodologies we used for improving the prediction of protein–ligand affinity; (a) the Virtual Screening method BINDSURF, and (b) two different softcomputing techniques are studied; neural networks (NN) and support vector machines (SVM) trained with different molecular properties calculated for known active and inactive compounds selected from standard VS benchmarks. In Fig. 1 a flowchart of the methodology is shown; once a protein target (component A) and a compound database (component B) have been chosen, compounds for which no information about affinity against protein target is available (component C) are docked using BINDSURF (component D) and estimated affinities (component E) and 3D poses (component F) are obtained. Using the methods described in this section, we start selecting compounds from the database for which affinity data is available (component G), so that we can calculate relevant descriptors (component H) and train adequately neural networks and support vector machines (component I) so that affinities obtained in component E are post-processed and we finally obtain improved values for the affinities (component J).

2.1. Virtual Screening with BINDSURF

The main idea underlying our VS method BINDSURF is the protein surface screening method, implemented in parallel on GPUs. Essentially, VS methods screen a large database of molecules in order to find which one fit some established criteria [14]. In the case of the discovery of new leads, compound optimization, toxicity evaluation and additional stages of the drug discovery process, we screen a large compound database to find a small molecule which interacts in a desired way with one or many different receptors. Among the many available VS methods for this purpose we decided to use protein–ligand docking [15,16]. These methods try to obtain rapid and accurate predictions of the 3D conformation a ligand adopts when it interacts with a given protein target, and also the strength of this union, in terms of its scoring function value. Docking simulations are typically carried out using a very concrete part of the protein surface in methods like Autodock [17], Glide [18] and DOCK [19], to name a few. This region is commonly derived from the position of a particular ligand in the crystal structure, or from the crystal structure of the protein without any ligand. The former can be performed when the protein is co-crystallized with the ligand, but it might happen that no crystal structure of this ligand–protein pair is at disposal. Nevertheless, the main problem is to take the assumption, once the binding site is specified, that many different ligands will interact with the protein in the same region, discarding completely the other areas of the protein.

Given this problem we propose to overcome it by dividing the whole protein surface into defined regions. Next, docking simulations for each ligand are performed simultaneously in all the specified protein spots. Following this approach, new hotspots might be found after the examination of the distribution of scoring function values over the entire protein surface. This information could lead to the discovery of novel binding sites. If we compare this approach with a typical docking simulation performed only in a region of the surface, the main drawback of this approach lies on its increased computational cost. We decided to pursue in this direction and show how this limitation can be overcome thanks to GPU hardware and new algorithmic designs.

In essence, in a docking simulation we calculate the ligand–protein interaction energy for a given starting configuration of the system, which is represented by a scoring function [20]. In BINDSURF the scoring function calculates electrostatic (ES), Van der Waals (VDW) and hydrogen bond (HBOND) terms.

Furthermore, in docking methods it is normally assumed [14] that the minima of the scoring function, among all ligand–protein conformations, will accurately represent the conformation the system adopts when the ligand binds to the protein. Thus, when the simulation starts, we try to minimize the value of the scoring function by continuously performing random or predefined perturbations of the system, calculating for each step the new value of the scoring function, and accepting it or not following different approaches like the Monte Carlo minimization method [21] or others. Simulations were always carried out with a total of 500 Monte

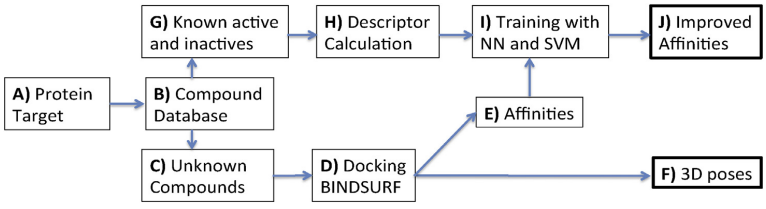


Fig. 1. Flowchart of the methodology used for improving the predictive capability of BINDSURF.

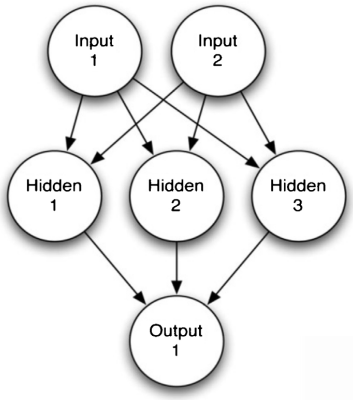


Fig. 2. Single hidden layer neural network.

Carlo steps. For a detailed discussion it is advisable to have a look at our previous BINSURF publication [12].

2.2. Softcomputing methods

We review the softcomputing methods we will apply to refine the prediction capacities of BINDSURF.

2.2.1. Neural networks

One of the most dominant application areas of neural networks is non-linear function approximation. The main advantage of neural network modeling is that complex non-linear relationships can be modeled without assumptions about the form of the model. That feature is very useful in the field of drug design and discovery.

In the last years a large number of authors have designed hybrid methods that combined neural networks with other techniques to solve chemistry related problems.

More than two decades ago, the aqueous solubility of organic compounds was studied using neural approaches [22]. In next decade, supervised and unsupervised neural models were employed to model QSAR, predict molecules activities and structure, clustering and many more [23,24]. More recently the problem of drug solubility prediction from structure has been revisited [25]. The prediction of physico-chemical properties of organic compounds from molecular structure has been extensively studied using hybrid techniques that include neural networks [26,27,28]. Also identification of small-molecule ligands has been improved using neural techniques [29,30,31].

There are several types of feed-forward neural networks (NNET), the most widely used being multi-layer networks with sigmoidal activation functions (multi-layer perceptrons) and single layer networks with local activation functions (radial basis function networks). The good approximation capability of neural networks has been widely demonstrated by both practical applications and theoretical research. We decided to use a single-hidden-layer neural network with skip-layer connections in this study (see Fig. 2) since it has been clearly demonstrated its impact on the differentiation between active and inactive compounds and other chemical applications [24]. For such purpose we used the *nnet* function of the R package [32].

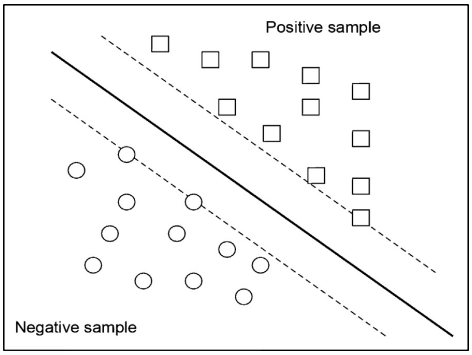


Fig. 3. Support vector machines margin hyperplanes.

2.2.2. Support vector machines

Support vector machines (SVM) [33] are a group of supervised learning methods that can be applied to classification or regression. They represent the decision boundary in terms of a typically small subset of all training examples, called the support vectors. In a short period of time, SVM have found numerous applications in chemistry, such as in drug design (discriminating between ligands and nonligands, inhibitors and noninhibitors, etc.) [34], drug discovery [35], quantitative structure–activity relationships (QSAR, where SVM regression is used to predict various physical, chemical, or biological properties) [36], chemometrics (optimization of chromatographic separation or compound concentration prediction from spectral data as examples), and sensors (for qualitative and quantitative prediction from sensor data), chemical engineering (fault detection and modeling of industrial processes) [37]. An excellent review of SVM applications in chemistry was published by Ivancicuc [38].

In our case, we exploit the idea that SVM produce a particular hyperplane in feature space, that separates active from inactive compounds, called the maximum margin hyperplane (see Fig. 3).

Most used kernels within SVM include: linear (dot), Polynomial, Neural (sigmoid,Tanh), Anova, Fourier, Spline, B Spline, Additive, Tensor and Gaussian Radial Basis or Exponential Radial Basis.

2.3. Ligand databases and molecular properties

We carried out our study applying the methods described in Sections 2.2.1 and 2.2.2 and using different sets of molecules that are known to be active or inactive. We employed standard VS benchmark tests, such as the Directory of Useful Decoys (DUD) [39], where VS methods check how efficient they are in differentiating ligands that are known to bind to a given target, from non-binders or decoys. Input data for each molecule of each set contains information about its molecular structure and whether it is active or not. We focused on three diverse DUD datasets (details are shown in Table 1) that cover kinases, nuclear hormone receptors and other enzymes such as TK, which corresponds to thymidine kinase (from PDB 1KIM), MR, which corresponds to mineralocorticoid

Table 1
Number of active (ligands) and inactive compounds (decoys) for each of the ligand datasets used in this study and obtained from DUD.

Protein	PDB Code	Resolution (Å)	No. of ligands	No. of decoys
GPB	1a8i	1.8	52	1851
MR	2aa2	1.9	15	535
TK	1kim	2.1	22	785

receptor (from PDB 2AA2), and GPB, which corresponds to the enzyme glycogen phosphorylase (from PDB 1A8I).

Next, using the ChemoPy package [40] we calculated for all ligands of the TK, MR and GPB sets a diverse of molecular properties derived from the set of constitutional, CPSA (charged partial surface area) and fragment/fingerprint-based descriptors, as described in Table 2. Constitutional properties depend on very simple descriptors of the molecule that can be easily calculated just counting the number of molecular elements such as atoms, types of atoms, bonds, rings, etc. These descriptors should be able to differentiate very dissimilar molecules, but might have problem for separating closely related isomers. CPSA descriptors take into account finer details of molecular structure, so they might be able to separate similar molecules, but might also have also difficulties for separating isomers. Lastly, fragment and fingerprint-based descriptors take into account the presence of an exact structure (not a substructure) with limited specified attachment points. These descriptors are more difficult to calculate. In generating the fingerprints, the program assigns an initial code to each atom. The initial atom code is derived from the number of connections to the atom, the element type, atomic charge, and atomic mass. This corresponds to an ECFP with a neighborhood size of zero. These atom codes are then updated in an iterative manner to reflect the codes of each atoms neighbors. In the next iteration, a hashing scheme is employed to

Table 2
Molecular descriptors used in this study.

Constitutional Descriptors	
Natom	Number of atoms
MolWe	Molecular Weight
NRing	Number of rings
NArRg	Numer of aromatic rings
NRotB	Number of rotatable bonds
NHDon	Number of H-bond donors
NHAcc	Number of H-bond acceptors
CPSA Descriptors	
Msurf	Molecular surface area
Mpola	Molecular polar surface area
Msolu	Molecular solubility
AlogP	Partition coefficient
Fragment/fingerprint-based descriptors	
ECP2, ECP4, ECP6	Extended-connectivity fingerprints (ECFP)
EstCt	Estate counts
AlCnt	AlogP2 estate counts
EstKy	Estate keys
MDLPK	MDL public keys

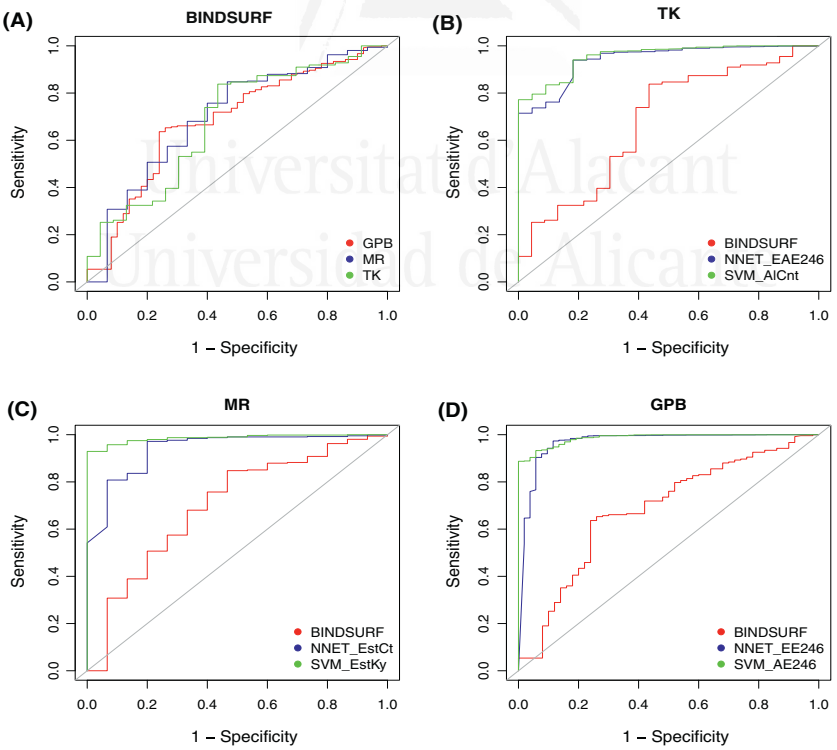


Fig. 4. From up-left to down right ROC plots; (A) obtained using BINDSURF for the targets of the DUD data set GPB (red), MR (blue) and TK (green), (B) obtained for data set TK using BINDSURF (red), the property that yields top AUC value (EAE246) using NNET, and the property that yields top AUC value (AlCnt) using SVM, (C) obtained for data set MR using BINDSURF (red), the property that yields top AUC value (EstCt) using NNET, and the property that yields top AUC value (EstKy) using SVM, and (D) obtained for data set GPB using BINDSURF (red), the property that yields top AUC value (EE246) using NNET, and the property that yields top AUC value (AE246) using SVM. In all cases diagonal line indicates random performance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Table 3
Obtained values for AUC of the ROC curves for the docking programs BINDSURF, DOCK, ICM and GLIDE when processing DUD datasets TK, MR and GPB.

Dataset	BINDSURF	DOCK	ICM	GLIDE
TK	0.700	0.521	0.723	0.681
MR	0.695	0.554	0.789	0.856
GPB	0.675	0.454	0.462	0.823

incorporate information from each atoms immediate neighbors. Each atoms new code now describes a molecular structure with a neighborhood size of one. This process is carried out for all atoms in the molecule. When the desired neighborhood size is reached, the process is complete and the set of all features is returned as the fingerprint. For the ECFPs employed in this paper, neighborhood sizes of two, four and six (ECFP 2, ECFP 4, ECFP 6) were used to generate the fingerprints. The resulting ECFPs can represent a much larger set of features than other fingerprints and contain a significant number of different structural units crucial for the molecular comparison, among the compounds.

3. Results and discussion

A set of experiments has been carried out in order to test the validity of our initial hypothesis combining and refining BINDSURF results with the proposed softcomputing methods.

3.1. Virtual Screening with BINDSURF

After BINDSURF calculations (depicted by components A, B, C and D in Fig. 1), results for three different DUD datasets (depicted by components E and F in Fig. 1) are shown in the ROC curves of Fig. 3. Given the results obtained for the DUD datasets TK, MR and GPB, and characterized by the value of the area under the curve (AUC) for each ROC curve, it could be said that, on average, BINDSURF performs similarly well than other docking methods such as DOCK [41], ICM [42] and GLIDE [18] as reported for these datasets [43] and shown in Table 3.

Nevertheless, it is clear that there is still room for improvement in the scoring function that BINDSURF uses, and on its energy optimization method (Monte Carlo), since both affect directly to the effectiveness of the direct prediction of binding poses.

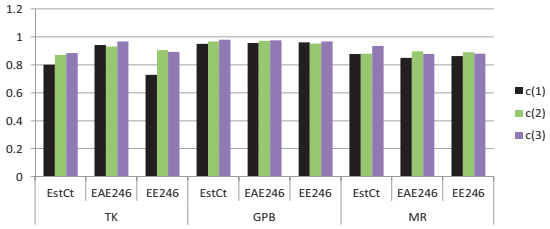


Fig. 5. AUC values of the ROC curves for some of the properties calculated in this study for the different datasets TK, GPB and MR, and for different numbers of neurons per layer, c(1), c(2) and c(3) used in NNET.

3.2. Activity prediction using softcomputing methods

NNET and SVM were trained with the previously described DUD datasets TK, MR and GPB (depicted by component G in Fig. 1). Molecular properties described in Table 2 were calculated for each molecule (depicted by components H in Fig. 1) as described in the methods section. A k-folds cross-validation technique with $k = 5$ was employed for neural networks and SVM experiments (depicted by component I in Fig. 1).

3.2.1. Neural networks

A set of experiments was carried out in order to find the most optimal feedforward neural network architecture for the classification problem proposed. Different numbers of neurons for the hidden layer were tested with the different descriptors and datasets previously described. Best results were obtained with 3 neurons per layer, c(3), for most of the properties and datasets tested. Increasing the number of neurons did not improve the results, as shown in Fig. 5.

After having chosen 3 neurons per layer for the NNET architecture, we can observe in Fig. 6 results obtained for all properties and different datasets, previously described.

3.2.2. Support vector machines

A set of experiments with different kernels was carried out in order to find the option with the best discrimination capacity between active and non-active compounds for each descriptor. More specifically, linear, polynomial, sigmoidal and radial kernels were tested with all the descriptors and datasets, and best results were obtained with radial kernel. The results obtained for AUC

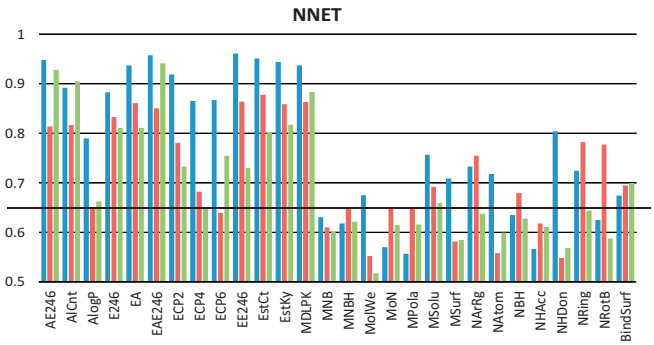


Fig. 6. AUC values of the ROC curves obtained using NNET as described in Section 2.2.1 for each property of Table 2 of the three different datasets GPB (blue), MR (red) and TK (yellow). Baseline for AUC = 0.65 is also shown. The resulting AUC values for the combined properties described in Table 5 are also reported, and also AUC values obtained by BINDSURF as depicted in Fig. 4A. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

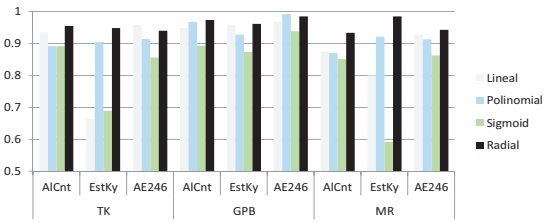


Fig. 7. AUC values of the ROC curves of some relevant properties obtained using SVM for different kernels.

values of the most relevant properties for different datasets and with different kernels are reported in Fig. 7.

We can observe in Fig. 8 results obtained with SVM and radial kernel for all properties and different datasets, previously described.

4. Discussion

BINDSURF predictions reach AUC values around 0.7 as reported in Fig. 4 and Table 3. From the other side, AUC values reported by both NNET and SVM depend clearly on the considered molecular property, and to a lesser extent, on the molecular dataset studied (GPB, MR, TK). The reason for the latter might be that main active compounds of these sets have similar structures (as shown in Fig. 9) consisting in small molecules with two or four rings, and also because they establish similar interactions with the protein, mainly based on hydrogen bond networks.

We propose a threshold value of 0.65 for AUC in order to discriminate which properties are useful for active/inactive prediction. Properties that yield simultaneously AUC values higher than this threshold for all sets using both NNET and SVM are; AlCnt, E246, ECP2 and MDLPK, while properties that yield AUC values lower than threshold are mostly AlogP, MolWe, MPola, MSolu, MSurf, NArgRg, Natom, NHacc, NHDon, NRing, and NRotB. So it seems clear that the best option for discriminating among active and inactive compounds in these datasets is to use fingerprint-based descriptors and to avoid the use of constitutional and CPSA descriptors. This is reasonable since fingerprint descriptors take into account more details about the structure of molecules, being able to efficiently discriminate with more accuracy between active compounds and their decoys.

Next, we studied whether combination of properties could lead to improvements on the predictive capability of these softcomputing methods. Therefore we combined properties that yielded the lowest AUC values, constitutional descriptors, and the properties that yielded the highest AUC values, so fingerprint based descriptors. Combinations used are described in Table 5 and AUC values obtained are reported in Figs. 6 and 8. In the case of combinations of constitutional descriptors, there is no clear improvement for either NNET or SVM, while for fingerprint combinations, average AUC values for the three datasets improve slightly.

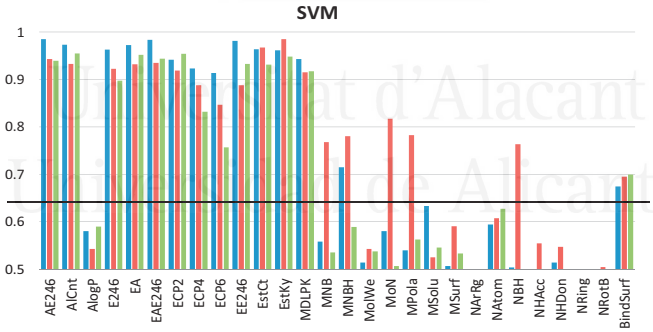


Fig. 8. AUC values of the ROC curves obtained using SVM as described in Section 2.2.1 for each property of Table 2 of the three different datasets GPB (blue), MR (red) and TK (yellow). Baseline for AUC = 0.65 is also shown. The resulting AUC values for the combined properties described in Table 5 are also reported, and also AUC values obtained by BINDSURF as depicted in Fig. 4A. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

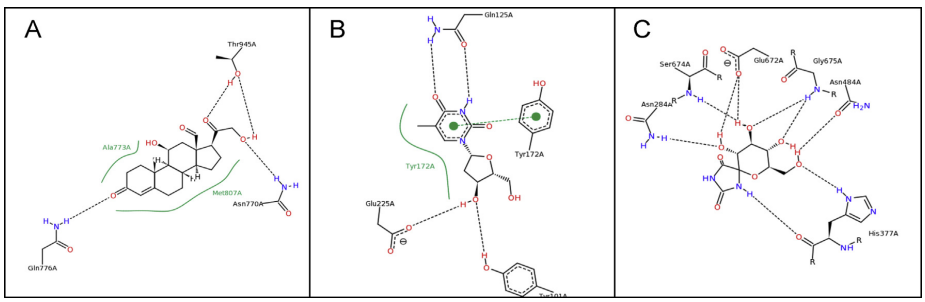


Fig. 9. Depiction of the molecular structure and protein–ligand interactions established by main active compounds from (A) MR, (B) TK, and (C) GPB.

Table 4
Top obtained values for AUC of the ROC curves when processing DUD datasets TK, MR and GPB for BINDSURF, NNET and SVM. For each dataset, the property that yields that top value of AUC for both NNET and SVM is specified.

TK		MR		GPB	
NNET_EE246	0.94	NNET_EstCt	0.87	NNET_EAE246	0.96
SVM_AE246	0.95	SVM_EstKy	0.98	SVM_AlCnt	0.98
BINDSURF	0.70	BINDSURF	0.70	BINDSURF	0.68

Table 5
Combinations of molecular descriptors used in this study.

Combinations of constitutional descriptors	
MNBH	Molecular polar surface area (MPola)+ Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MNB	Molecular polar surface area (MPola) + Number of rotatable bonds (NRotB)
NBH	Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
Combinations of fragment/fingerprint-based descriptors	
EAE246	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EA	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt)
AE246	AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EE246	Estate counts (EstCt) + Extended-connectivity fingerprints (ECFP)

Finally, top obtained AUC values for datasets GPB, MR and TK correspond to properties EE246 (0.96), EstCt (0.87) and EAE246 (0.94) when using NNET, and AE246 (0.98), EstKy (0.98) and AlCnt (0.95) when using SVM. Obtained ROC curves for the mentioned top properties can be seen in Fig. 4. Therefore, if we compare NN and SVM obtained results with the previous ones obtained by BINDSURF, as reported in Fig. 4B–D and Table 4, it is clear that predictive capability increases when using the presented methodology with the application of softcomputing methods as depicted in Fig. 1.

Consequently, and taking into account information obtained by softcomputing methods, we can post-process docking results obtained by the scoring function of BINDSURF (as depicted in the flowchart in Fig. 1) and neglect resulting compounds that are predicted as inactive. Then we can sort them by the final affinity value predicted by the BINDSURF scoring function for such cases and study visually the top ones. As an example we can observe in Fig. 10 the good agreement in the comparison between the top predicted compound from the DUD database for GPB and the crystallographic pose. In this case main stabilizing interactions are due to a hydrogen-bond network where the intervening nitrogen and oxygen atoms of the predicted compound fall closely to the same atoms of the crystallographic pose.

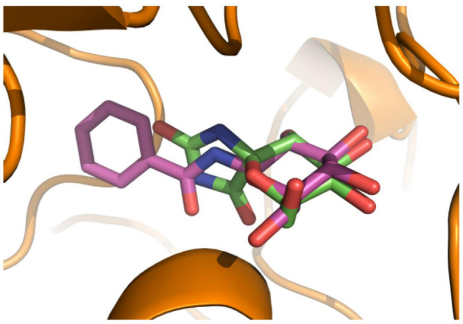


Fig. 10. Depiction of the binding mode found by BINDSURF for ligand number 17 of the DUD data set of GPB (PDB ID: 1A8I) with pink skeleton and crystallographic pose for Beta-D-Glucopyranose Spirohydantoin, with green skeleton. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

5. Conclusions

In this work we have shown how the predictive capability of the VS method BINDSURF can be improved applying softcomputing methods such as neural networks and support vector machines when using only a small set of representative chemical properties. We have also studied which of these properties are the most representative, and we have finally obtained that topological properties can efficiently discriminate between active and non-active compounds for the datasets studied. However, it must be mentioned that softcomputing approaches can only be used when there is data available for active and non-active compounds for a given protein. For further studies we consider it would be of high interest to train softcomputing methods with a diverse range of absolute affinity data for known compounds and to check whether prediction accuracy still increases with respect to the methodology presented on this work.

Given the improvements shown in the obtained results, we conclude that this methodology can be used to improve drug discovery, drug design, repurposing and therefore aid considerably in biomedical research. In the next steps we want to substitute the Monte Carlo minimization algorithm already present in BINDSURF for more efficient optimization alternatives, such as the Ant Colony optimization method, which we have already efficiently implemented on GPU [44] and implement also full ligand and receptor flexibility.

Acknowledgements

We thank the Catholic University of Murcia (UCAM) under grant PMAFI/26/12. This work was partially supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain. The authors also thankfully acknowledge the computer resources and the technical support provided by the Plataforma Andaluza de Bioinformática of the University of Málaga. We would also like to thank Dr. Andrés Bueno-Crespo for fruitful discussions. Experiments were also made possible thanks to a generous donation of hardware from NVDIA.

References

[1] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, Nucleic Acids Res. 28 (2000) 235–242.
[2] R. Sanchez, A. Sali, Large-scale protein structure modeling of the *Saccharomyces cerevisiae* genome, Proc. Natl. Acad. Sci. U. S. A. 95 (23) (1998) 13597–13602.
[3] L. Yao, J.A. Evans, A. Rzhetsky, Novel opportunities for computational biology and sociology in drug discovery, Trends Biotechnol. 27 (2009) 531–540.
[4] M. Garland, D.B. Kirk, Understanding throughput-oriented architectures, Commun. ACM 53 (2010) 58–66.
[5] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, V. Volkov, Parallel computing experiences with cuda, IEEE Micro 28 (2008) 13–27.
[6] NVIDIA, Whitepaper NVIDIA's Next Generation CUDA Compute Architecture: Fermi, 2009.
[7] H. Pérez-Sánchez, W. Wenzel, Optimization methods for virtual screening on novel computational architectures, Curr. Comput. Aided Drug Des. 7 (1) (2011) 44–52.
[8] G. Guerrero, H. Pérez-Sánchez, W. Wenzel, J.M. Cecilia, J.M. García, Effective parallelization of non-bonded interactions kernel for virtual screening on gpus, in:

5th International Conference on Practical Applications of Computational Biology; Bioinformatics (PACBB 2011), vol. 93, Springer, Berlin/Heidelberg, 2011, pp. 63–69.
[9] I. Sánchez-Linares, H. Pérez-Sánchez, G.D. Guerrero, J.M. Cecilia, J.M. García, Accelerating multiple target drug screening on gpus, in: Proceedings of the 9th International Conference on Computational Methods in Systems Biology (CMSB'11), ACM, New York, NY, USA, 2011, pp. 95–102.
[10] I. Sánchez-Linares, H. Pérez-Sánchez, J.M. García, Accelerating grid kernels for virtual screening on graphics processing units, in: E. D'Hollander, D. Padua (Eds.), Parallel Computing: Proceedings of the International Conference ParCo, vol. 22, IOS, 2011, pp. 413–420.
[11] G. Brannigan, D.N. LeBard, J. H_enin, R.G. Eckenho_, M.L. Klein, Multiple binding sites for the general anesthetic isourane identified in the nicotinic acetylcholine receptor transmembrane domain, Proc. Natl. Acad. Sci. U. S. A. 107 (32) (2010) 14122–14127.
[12] I. Sánchez-Linares, H. Pérez-Sánchez, J. Cecilia, J. García, High-throughput parallel blind virtual screening using bindsurf, BMC Bioinformatics 13 (Suppl. 14) (2012) S13.
[13] C. Hetényi, D. van der Spoel, Efficient docking of peptides to proteins without prior knowledge of the binding site, Protein Sci. 11 (7) (2002) 1729–1737.
[14] W. Jorgensen, The many roles of computation in drug discovery, Science 303 (5665) (2004) 1813–1818.
[15] E. Yuriev, M. Agostino, P.A. Ramsland, Challenges and advances in computational docking: 2009 in review, J. Mol. Recogn. 24 (2) (2011) 149–164.
[16] S.Y. Huang, X. Zou, Advances and challenges in protein–ligand docking, Int. J. Mol. Sci. 11 (8) (2010) 3016–3034.
[17] G. Morris, D. Goodsell, R. Halliday, R. Huey, W. Hart, R. Belew, A. Olson, Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function, J. Comput. Chem. 19 (14) (1998) 1639–1662.
[18] R.A. Friesner, J.L. Banks, R.B. Murphy, T.A. Halgren, J.J. Klicic, D.T. Mainz, M.P. Repasky, E.H. Knoll, M. Shelley, J.K. Perry, et al., Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy, J. Med. Chem. 47 (7) (2004) 1739–1749.
[19] T.J.A. Ewing, S. Makino, A.G. Skillman, I.D. Kuntz, Dock 4.0. Search strategies for automated molecular docking of exible molecule databases, J. Comput. Aided Mol. Des. 15 (5) (2001) 411–428.
[20] R. Wang, Y. Lu, X. Fang, S. Wang, An extensive test of 14 scoring functions using the PDBbind re_ned set of 800 protein–ligand complexes, J. Chem. Inform. Comput. Sci. 44 (6) (2004) 2114–2125.
[21] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (1953) 1087–1092.
[22] N. Bodor, A. Harget, M.J. Huang, Neural network studies. 1. Estimation of the aqueous solubility of organic compounds, J. Am. Chem. Soc. 113 (25) (1991) 9480–9483.
[23] K.L. Peterson, Artificial neural networks and their use in chemistry, Rev. Comput. Chem. 16 (2007).
[24] G. Schneider, P. Wrede, Artificial neural networks for computer-based molecular design, Prog. Biophys. Mol. Biol. 70 (3) (1998) 175–222.
[25] W.L. Jorgensen, E.M. Duffy, Prediction of drug solubility from structure, Adv. Drug Deliv. Rev. 54 (3) (2002) 355–366.
[26] J. Taskinen, J. Yliruusi, Prediction of physicochemical properties based on neural network modelling, Adv. Drug Deliv. Rev. 55 (9) (2003) 1163–1183.
[27] M. Weisel, J.M. Kriegl, G. Schneider, Architectural repertoire of ligand-binding pockets on protein surfaces, ChemBioChem 11 (4) (2010) 556–563.
[28] N.R. Pal, R. Panja, Finding short structural motifs for re-construction of proteins 3D structure, Appl. Soft Comput. 13 (2) (2013) 1214–1221.
[29] J.D. Durrant, J.A. McCammon, NNScore: a neural-network-based scoring function for the characterization of protein–ligand complexes, J. Chem. Inf. Model 50 (10) (2010) 1865–1871.
[30] J.D. Durrant, J.A. McCammon, NNScore 2.0. A neural-network receptor – ligand scoring function, J. Chem. Inf. Model 51 (11) (2011) 2897–2903.
[31] I.V. Romero Reyes, I.V. Fedyushkina, V.S. Skvortsov, D.A. Filimonov, Prediction of progesterone receptor inhibition by high-performance neural network algorithm, Int. J. Math. Model. Meth. Appl. Sci. 7 (3) (2013) 303–310.
[32] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S, 4th ed., Springer, New York, 2002, ISBN:0-387-95457-0.
[33] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[34] R.N. Jorissen, M.K. Gilson, Virtual screening of molecular databases using a support vector machine, J. Chem. Inf. Model 45 (2005) 549–561.
[35] M.K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, C. Lemmen, Active learning with support vector machines in the drug discovery process, J. Chem. Inf. Comput. Sci. 43 (2) (2003) 667–673.
[36] J.M. Kriegl, T. Arnhold, B. Beck, T. Fox, Prediction of human cytochrome P450 inhibition using support vector machines, QSAR Comb. Sci. 24 (2005) 491–502.
[37] D.E. Lee, J.H. Song, S.O. Song, E.S. Yoon, Weighted support vector machine for quality estimation in the polymerization process, Ind. Eng. Chem. Res. 44 (2005) 2101–2105.
[38] O. Ivanciuc, Applications of support vector machines, Chem. Rev. Comput. Chem. 23 (2007) 291–400.
[39] N. Huang, B.K. Shoichet, J.J. Irwin, Benchmarking sets for molecular docking, J. Med. Chem. 49 (23) (2006) 6789–6801.
[40] D.-S. Cao, Q.-S. Xu, Q.-N. Hu, Y.-Z. Liang, ChemoPy: freely available python package for computational biology and chemoinformatics, Bioinformatics (2013), http://dx.doi.org/10.1093/bioinformatics/btt105.
[41] B.K. Shoichet, D.L. Bodian, I.D. Kuntz, Molecular docking using shape descriptors, J. Comput. Chem. 13 (1992) 380–397.
[42] R. Abagyan, M. Totrov, D. Kuznetsov, ICM – a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation, J. Comput. Chem. 15 (1994) 488–506.
[43] J.B. Cross, D.C. Thompson, B.K. Rai, J.C. Baber, K.Y. Fan, Y. Hu, C. Humblet, Comparison of several molecular docking programs: pose prediction and virtual screening accuracy, J. Chem. Inf. Model 49 (6) (2009) 1455–1474.
[44] J.M. Cecilia, J.M. García, M. Ujaldon, A. Nisbet, M. Amos, Parallelization strategies for ant colony optimisation on gpus, in: 14th Int. Workshop on Nature Inspired Distributed Computing -NIDISC11- (in conjunction with IPDPS 2011), IEEE, 2011, pp. 339–346.

3 CONCLUSIONES Y CONTRIBUCIONES

En este capitulo se extraen las principales conclusiones de este trabajo de investigación. Está organizado en cuatro secciones: la sección 3.1 presenta y discute las conclusiones finales de los trabajos presentados en esta tesis. En la sección 3.2 se enumera las contribuciones más relevantes de este trabajo, mientras que la sección 3.3 lista las publicaciones derivadas del presente trabajo. Finalmente, la sección 3.4 presenta los trabajos y líneas de investitación futuras: problemas abiertos y temas pendientes para tratar en futuras investigaciones.

3.1 CONCLUSIONES

En este trabajo se ha demostrado que los métodos de inteligencia computacional como: las redes neuronales, las máquinas de soporte vectorial o los bosques aleatorios, pueden refinar la capacidad de predicción de los métodos de cribado virtual.

Sin embargo, se debe mencionar que los métodos de inteligencia computacional sólo se pueden usar cuando hay datos disponibles de compuestos activos e inactivos para una proteína dada, para los casos de bioactivad, y del mismo modo solo pueden predecir la solubilidad cuando hay datos disponibles de compuestos solubles e insolubles y así poder entrenar el modelo.

Las investigaciones en la mejora del rendimiento de los métodos de CV son de gran interés desde el punto de vista tecnológico. El refinamiento

de estos métodos mediante métodos de inteligencia computacional y la búsqueda automática de descriptores moleculares abre la puerta a la utilización de grandes bases de datos de compuestos bioactivos extensos, donde la naturaleza de los datos puede condicionar la selección de los descriptores a utilizar, consiguiendo con estas técnicas una selección automática y evitando así una selección manual (ad hoc) no óptima.

Se ha demostrado, de igual modo, que es posible utilizar GPUs de bajo presupuesto en lugar de caros supercomputadores, que permiten realizar cálculos computacionales intensivos, imposibles de llevar a cabo en el pasado, y que se vuelven ahora factibles. Este incremento de velocidad en las predicciones es considerable cuando tratamos con un gra número de moléculas, o con alta complejidad en los descriptores moleculares usados.

El conocimiento adquirido puede ser transferido a otros investigadores para que trasladen sus programas y aplicaciones a la próxima generación de arquitecturas computacionales masivamente paralelas y a nuevos métodos de inteligencia computacional.

Teniendo en cuenta las mejoras que se muestran en los resultados obtenidos, se concluye que esta metodología se puede utilizar para mejorar el descubrimiento, diseño y reutilización de fármacos y por lo tanto ayudar en el avance de la investigación biomédica.

3.2 CONTRIBUCIONES

Las principales contribuciones obtenidas, como resultado de este trabajo de investigación, son:

1. Se ha mejorado la capacidad de predicción de los métodos de CV mediante el refinamiento de resultados con la aplicación de métodos inteligencia computacional como:
    a. Redes Neuronales (NNET).
    b. Maquinas de Soporte Vectorial (SVM).
    c. Bosques Aleatorios (RF).
2. Se ha demostrado, así mismo, que un pequeño conjunto de propiedades químicas representativas pueden discriminar eficazmente entre los compuestos activos y no activos de los conjuntos de datos estudiados. En particular, La selección automática de descriptores moleculares mediante la técnica de RF mejora la predicción frente a la selección manual de descriptores ("Ad hoc").
3. Se ha rediseñado y paralelizado el método de inteligencia computacional basado en máquinas de soporte vectorial para aprovechar la capacidad de procesamiento de las GPU, reduciendo el tiempo de la predicción para el cálculo de una característica tan importante como la solubilidad. Testeando la propuesta sobre una base de datos con un gran número de moléculas y/o la utilización de descriptores moleculares complejos y consiguiendo aceleraciones de hasta 15x.

3.3 PUBLICACIONES

Las siguientes artículos fueron publicados como resultado de las investigaciones de esta tesis:

- Artículos publicados en *revistas internacionales con impacto:*
  - Horacio Pérez-Sánchez, Gaspar Cano and José García-Rodríguez: **Improving Drug Discovery using Hybrid Softcomputing Methods**. Applied Soft Computing 20 (2014) 119-126. JCR IMPACT FACTOR: 2,140. http://dx.doi.org/doi:10.1016/j.asoc.2013.10.033.

  - Gaspar Cano, José García-Rodríguez and Horacio Pérez-Sánchez: **Improvement of Virtual Screening predictions using Computational Intelligence methods**. Letters in Drug Design & Discovery, 11, 33-39, 2014. JCR IMPACT FACTOR: 0,845.

  - Horacio Pérez, Gaspar Cano, José García y José María Cecilia: **Descubrimiento de Fármacos con Cribado Virtual Refinado con Enfoques Neuronales Paralelos**. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería (RIMNI) 2014. JCR IMPACT FACTOR: 0,184. http://dx.doi.org/doi: 10.1016/j.rimni.2014.06.004.

- Artículos publicados en revistas internacionales:
  - Pérez-Sánchez H, Cecilia JM, Imbernón-Tudela B, Pérez-Garrido A, Soto-Iniesta J, Timón-Pérez I, García-Rodríguez J, Cano G, Bueno-Crespo A and Vegara-Meseguer: **The Need for an Integrated Computational/ Experimental Approach in the Discovery and Design of New Drugs.** J. Drug Des 2014, 3:1. http://dx.doi.org/10.4172/2169-0138.1000e121.

- Artículos presentados en Conferencias Internacionales:
  - Horacio Pérez-Sánchez, Ginés D. Guerrero, José M. García, Jorge Peña, José M. Cecilia, Gaspar Cano, Sergio Orts-Escolano, and José García-Rodríguez: **Improving Drug Discovery Using a Neural Networks Based Parallel Scoring Function**. International Joint Conference on Artificial Neural Networks. IJCNN, Dallas (USA), August 2013.

  - Cano, G., Botía Blaya, J, Palama, J,García-Rodríguez.;Pérez-Sánchez,h. **Improvement of virtual screening predictions using support vector machines** H. International Conference Computational and Mathematical Methods in Science and Engineering 2013. Almeria (Spain), Junio 2013.
    ISBN: 978-84-616-2723-3

3.4 TRABAJOS FUTUROS

Entre los trabajos y líneas de investitación futuras, resumimos en este apartado algunas de las más prometedoras:

1. En futuros estudios, sería de gran interés la utilización de otros métodos de inteligencia computacional diferentes a los aquí expuestos para la clasificación (predicción).
2. Para la selección automática de descriptores moleculares se ha utilizado RF pero seria muy interesante probar otros métodos SVM o C 5.0.
3. Existen otras arquitecturas GPU sobre las que se podría estudiar el incremento en prestaciones, para el procesamiento de grandes volúmenes datos.
4. En los ultimos años, Intel ha desarrollado una nueva arquitectura masivamente paralela, XeonPhi [141], [142]. Se plantea continuar el estudio y la comparación de ambas arquitecturas para demostrar las ventajas e inconvenientes de cada una y elegir la más adecuada para nuestro propósito.
5. *MapReduce* [143], es un modelo de programación, que da soporte a los métodos de inteligencia computacional ya descritos [144] , pero sobre grades bases de datos, *big data* [145] y entornos altamente distribuidos. Usu so podría aportar grandes ventajas y permitiría portar los resultados obtenidos a otros campos.
6. La búsqueda de candidatos para el refinamiento de los métodos de VS se hizo mediante la predicción de actividad y solubilidad, apoyada en el uso de descriptores moleculares. Esta selección de candidatos se puede enriquecer mediante el empleo de técnicas que explotan la similitud estructural (QSAR).

7. Finalmente, se plantea el descubrimiento de compuestos bioactivos en otros contextos de relevancia biológica. Actualmente, trabajamos en el proyecto NILS en colaboración con la University of Iceland y el Nordic Center of High Performance Computing (http://nilsbiohpc.hi.is/), en el que se estudiaran técnicas aceleradas de RF y SVM aplicadas al descubrimiento de Anticoagulantes.

ANEXOS

Esta tesis presentada por compendio compila los resultados de la investigación realizada en los últimos 3 años que dieron como fruto, además de las aportaciones presentadas en el capítulo 2 y publicadas en revistas de impacto, estas otras publicaciones que han sido enviadas a revistas de prestigio y se encuentran en proceso de revisión :

- **Automatic molecular descriptors selection using Random Forest: Application to drug discovery.** Submitted to Machine Learning Journal.

- **Support Vector Machines prediction of drug solubility on GPUs.**Submitted to the International Journal of Parallel Programming.

A. AUTOMATIC MOLECULAR DESCRIPTORS SELECTION USING RANDOM FOREST: APPLICATION TO DRUG DISCOVERY

Submitted to Machine Learning Journal.

Resumen: La selección óptima de características (descriptores moleculares) es un paso previo fundamental para un uso eficiente de las técnicas de inteligencia computacional en aplicaciones de diseño y descubrimiento de fármacos. La selección adecuada de las propiedades moleculares y su naturaleza, condicionará el acierto en la predicción de la actividad. Con el fin de mejorar la predicción de la actividad se hace uso de bosques aleatorios (RF) para la selección de los descriptores moleculares. Dado que la validez del resultado obtenido, para un conjunto de datos, está condicionada por los descriptores utilizados, su selección es crucial para maximizar la bondad del ajuste y validar la robustez del planteamiento empleado. Una vez aplicado el proceso de selección de variables de interés, podremos entrenar de nuevo el modelo con las bases de datos de compuestos activos o inactivos conocidos. Esta información será utilizada para mejorar posteriormente las predicciones y en la aceleraración del descubrimiento de nuevos fármacos aplicando técnicas de cribado virtual.

Automatic Selection of Molecular Descriptors using Random Forest: Application to Drug Discovery

Gaspar Cano[1], Horacio Pérez-Sánchez[2] and José García-Rodríguez[1]

[1]Dept. of Computing Technology, University of Alicante, PO.Box. 99.
E03080.Alicante, Spain {jgarcia,gcano}@dtic.ua.es

[2] Computer Science Department, Catholic University of Murcia (UCAM)
E30107 Murcia, Spain {hperez}@ucam.edu

Abstract. The optimal selection of features (molecular descriptors) is an essential pre-processing step for the efficient use of computational intelligence techniques in the design of applications for drug discovery. Theselection of the molecular descriptors have key influence in the accuracy of the prediction of affinity. In order to improve this prediction, we propose a Random Forest (RF) based approach to improve the selection of molecular descriptors. The validity of the results obtained (bioactive or not) for a dataset is determined by the descriptors used. This selection is crucial to maximize the results and evaluate the robustness of the approach used. We also propose a hybrid novel approach where Random Forest methods are trained with databases of known active (drugs) or inactive compounds, being this information exploited afterwards to improve Virtual Screening predictions.

Keywords: Random Forest, Clinical Research, Drug Discovery, Virtual Screening.

1 Introduction

The improvement in the selection of candidates by screening large databases of compounds using computational intelligence techniques is essential in predicting the activity and this allows virtual screening (VS) techniques become more efficient. This a priori selection allows us to reduce the number of potential candidates that should be evaluated for suitability. Selection of characteristics (molecular descriptors) with greater discriminatory power, allows us to predict which compounds will be good candidates for later use in refining virtual screening methods [1], and in turn drastically reduce the computational complexity and time, noting the problem and allowing focus the computational effort of the candidates proposed as active in order to accelerate biomedical research.

The techniques of extraction and selection of features, in our domain, molecular descriptors, are intended to reduce the size of the dataset, considering the most relevant variables for each case, based on the characteristics available for each dataset.

To find these features, we propose a hybrid approach where computational intelligence methods such as Random Forest (RF), trained with databases of known active (drugs) and inactive compounds help to define the best descriptors that provide the most relevant information in the classification stages.

After the automatic selection of these molecular descriptors, we apply again this technique to determine the goodness of the selection to provide the prediction of its activity. Figure 1 shows data flow, feature selection, from dataset selection to obtain the best AUC (Area Under the Curve) for each dataset.

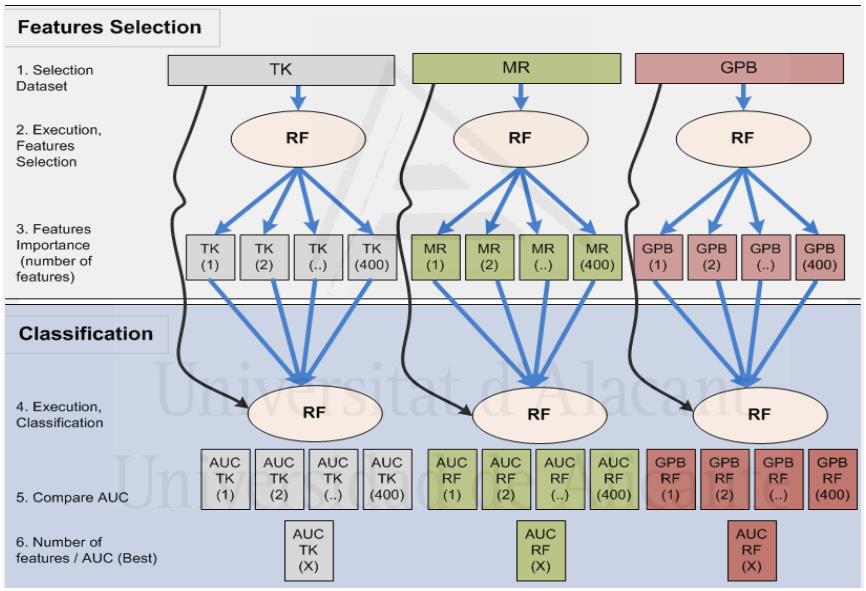


Fig. 1. Data flow for automatic feature selection.

The rest of the paper is organized as follows: section 2 describes the methodology used in the selection of variables and a method of computational intelligence is introduced (RF). In section 3, a set of experiments to fit and model the automatic feature selection are presented. Then, in Section 4 a discussion of the results is made and ends with the conclusions of the work and future lines below.

2. Methodology

This section describes the methods we used to improve the selection of molecular descriptors. Computational intelligence techniques such as Random Forest (RF) are used for the selection of molecular descriptors and this method is trained with different datasets that have been widely used by different VS techniques. Automatic selection of variables is compared with data obtained from the manual selection (ad hoc) of these same descriptors tested in a previous study [1].

2.1 Features Selection

For help the system to take decision about the classification, it is necessary to carefully select the predictor variables which must help to decide between input features chosen [2]. The set of features that describes an object can be arbitrarily large, so that in most cases a pre-selection stage is required.

The input variables (predictors) for a dataset are a fixed number of features, in our domain, the molecular descriptors. The values of these predictors can be binary, categorical or continuous and represent the set of the system input data.

The feature selection process consists of two main stages: acquisition of data (filtering, suitability, scaling) and feature selection. What are most relevant features for our application domain? As we are working with standardized databases, we avoid steps for filtering, scaling or deciding the suitability of this data. We will focus on the selection of features, there are different motivations for doing so, but we will seek to obtain a number of benefits [3]. In particular, we hope to get some of the following benefits:

- Reduction of the total number of data and reduction of global information.
- Reduction of features, reducing the cost of continued storage.
- Improved performance, improved processing speed can lead to an improvement in prediction accuracy.
- Improved display, improved representation helps the understanding of the problem.
- Reduced training time, smaller data subset decreases training time.
- Reduction of noise in the data, removing irrelevant or redundant features.

2.1.1 Automatic Selection of Descriptors

A proper selection of the set of molecular descriptors (predictors) is essential to optimize the prediction and automatic selection of these descriptors. This is a clear objective of automatic versus manual selection (ad hoc) methods. What are the most important variables in the classification models? This problem is common in many research domains. Usually, it is solved using the variable that

best explains our model and adapt to the domain in which we work. For some domains, the segmentations criteria are simple or are constructed around artificial variables (dummy). These are the mechanism that are adopted by a domain expert and sometimes is a multidisciplinary task, in any case built for each particular problem ("ad hoc"), we try to predict or classify. The uses of computational intelligence techniques allow us to select these variables in an automatic way by quantifying the relative importance of the variables.

2.1.2 Minimal subset of features

Once the idea of the relevance of the selected features is introduced, those not selected, or who have been left out, should be irrelevant or redundant. Therefore, the order of relevance allows us to extract a minimal subset of features that are enough to make an optimal prediction. In Random Forest, the classification method is based on the use of decision trees on multiple samples of a data set. The ability to include a large number of input variables in our model (predictors) to find linear relationships between them and avoid that may appear random, make this method very interesting for this purpose.

2.1.3 Ligand databases and molecular properties

We employed standard VS benchmark tests, such as the Directory of Useful Decoys (DUD) [4], where VS methods check how efficient they are in differentiating ligands that are known to bind to a given target, from non-binders or decoys. Input data for each molecule of each set contains information about its molecular structure and whether it is active or not. We focused on three diverse DUD datasets (details are shown in Table 1) that cover kinases, nuclear hormone receptors and other enzymes such as TK, which corresponds to thymidine kinase (from PDB 1KIM), MR, which corresponds to mineralocorticoid receptor (from PDB 2AA2), and GPB, which corresponds to the enzyme glycogen phosphorylase (from PDB 1A8I).

Table 1. Number of active (ligands) and inactive compounds (decoys) for each of the ligand datasets used in this study and obtained from DUD.

Protein	PDB Code	Resolution (Å)	n_o of Ligands	n_o of Decoys
GPB	1A8I	1.8	52	1851
MR	2AA2	1.9	15	535
TK	1KIM	2.1	22	785

Next, using the ChemoPy package [5] we calculated, for all ligands of the TK, MR and GPB sets, a seto of diverse of molecular properties derived from the set

of constitutional, CPSA (charged partial surface area) and fragment/fingerprint-based descriptors, as described in Table 2. Constitutional properties depend on very simple descriptors of the molecule that can be easily calculated just counting the number of molecular elements such as atoms, types of atoms, bonds, rings, etc. These descriptors should be able to differentiate very dissimilar molecules, but might have problem for separating closely related isomers. CPSA descriptors take into account finer details of molecular structure, so they might be able to separate similar molecules, but might also have also difficulties for separating isomers. Lastly, fragment and fingerprint-based descriptors take into account the presence of an exact structure (not a substructure) with limited specified attachment points. These descriptors are more difficult to calculate. In generating the fingerprints, the program assigns an initial code to each atom. The initial atom code is derived from the number of connections to the atom, the element type, atomic charge, and atomic mass. This corresponds to an ECFP with a neighbours size of zero. These atom codes are then updated in an iterative manner to reflect the codes of each atoms neighbour. In the next iteration, a hashing scheme is employed to incorporate information from each atoms immediate neighbour. Each atoms new code now describes a molecular structure with a neighbourhood size of one. This process is carried out for all atoms in the molecule. When the desired neighbourhood size is reached, the process is complete and the set of all features is returned as the fingerprint. For the ECFPs employed in this paper, neighbourhood sizes of two, four and six (ECFP 2, ECFP 4, ECFP 6) were used to generate the fingerprints.

Table 2. Molecular descriptors used in this study.

CONSTITUTIONAL DESCRIPTORS	
Natom	Number of atoms
MolWe	Molecular Weight
NRing	Number of rings
NArRg	Number of aromatic rings
NRotB	Number of rotatable bonds
NHDon	Number of H-bond donors
NHAcc	Number of H-bond acceptors
CPSA DESCRIPTORS	
Msurf	Molecular surface area
Mpola	Molecular polar surface area
Msolu	Molecular solubility
AlogP	Partition coefficient
FRAGMENT/FINGERPRINT-BASED DESCRIPTORS	
ECP2, ECP4, ECP6	Extended-connectivity fingerprints (ECFP)
EstCt	Estate counts
AlCnt	AlogP2 Estate counts
EstKy	Estate keys
MDLPK	MDL public keys

The resulting ECFPs can represent a much larger set of features than other fingerprints and contain a significant number of different structural units crucial for the molecular comparison, among the compounds. Table 3 shows the combinations of molecular descriptors used, combinations of constitutional descriptors and descriptors based on molecular fingerprints.

Table 3. Combinations of molecular descriptors used in this study.

COMBINATIONS OF CONSTITUTIONAL DESCRIPTORS	
MNBH	Molecular polar surface area (MPola)+ Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MNB	Molecular polar surface area (MPola) + Number of rotatable bonds (NRotB)
NBH	Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
COMBINATIONS OF FRAGMENT/FINGERPRINT-BASED DESCRIPTORS	
EAE246	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EA	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt)
AE246	AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EE246	Estate counts (EstCt) + Extended-connectivity fingerprints (ECFP)

2.2 Computational Intelligence Methods

The use of methods of computational intelligence will permit us to provide a sufficient subset of features. Since the early 50 computational intelligence research has focused on finding relationships between data and analyse these relationships [6]. These problems are found in a wide variety of application domains: engineering, robotics or pattern recognition [7]. Systems that recognize writing [8], voice [9], pictures [10], sequencing genes [11], diagnose illnesses [12], interpret traffic signs [13] or reject spam [14] are good examples.

Given a number of training data (training) associated with an expected output, the computational intelligence processes allow us to find the relationship between the pattern and the expected result, using these training data. The goal is to predict the unknown output, for a new dataset (test). The generalization of this task and building a predictive model or predictor, which contains some adjustable parameters, make up the model. Training data are used for the optimal selection

of these parameters, and different algorithms are used from a broad range of computational intelligence techniques. A classifier is a function that assigns to an unlabelled sample a label or class. A sample of several predefined categories or classes is classified. Classification models can be constructed using a variety of algorithms[15].

2.2.1 Random Forest.

Random Forest [16] (Figure 2) is a supervised learning method that can be applied to solve classification or regression problems. It is constituted by a combination of tree predictors such that each tree depends on the values of a random vector, independently and with the same layout for each of the generated vectors. The decision tree takes as a whole class that gets the most votes in the whole tree. Many disciplines use Random Forest: Accident analysis [17] , mechanical Engineering [18], financial engineering [19], [20], language models [21] o biology [22].

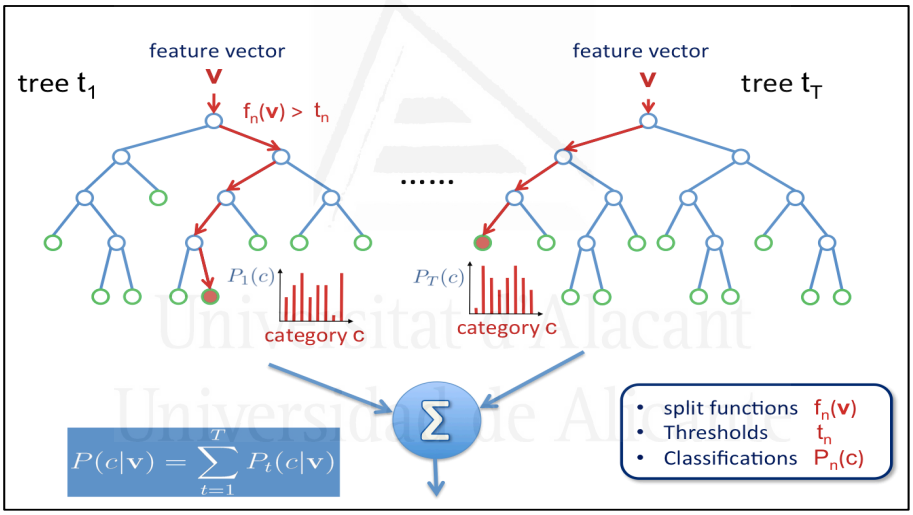


Fig. 2. Random Forest is "a collection of classifiers that are structured as trees t_n where $F_n(v)$ are independent and identically distributed random vectors and each tree produces a vote of the most popular class for an input x (predictor)". The random vectors $P_n(c)$ represents a set of random numbers that determine the construction of each tree.

In Random Forest [23] each individual tree is explored in a particular way :

1. Given a set of training data *N*, *n* random samples with repetition (Bootstrap) are taken as training set.
2. For each node of the tree, *M* input variables are determined, and "*m*" << *M*, variables are selected for each node. The most important variable randomly chosen is used as a node. The value of m remains constant during the expansion of forest.
3. Each tree is developed to its maximum expansion, never pruning.

The error of the set of trees depends on two factors:

- Correlation between any two trees in the forest, avoiding the use of a subset of variables randomly chosen data resampling (Bootstrap).
- A strong classifier, the importance of each tree in the forest, shows that with a low value of this error, the increase of these classifiers decreases the forest error.

2.2.2 Error estimation

To estimate the classification or regresion error in RF [6], it is defined the OOB (out-of-bag), that simple estimate a selection of the input observations based on Bagging [24], (resampling of a random subset of predictors to be replaced in each tree). On average, each tree Bagging, uses two-thirds of the observations, the remaining third will not be used in the comments off-exchange (OOB). So you can predict the response to the i-nth observation using each tree will produce B / 3 predictions for observation i. In order to obtain a single prediction for the ith element, we forecast average these responses (for regression) or by majority vote (for classification). OOB there is one prediction for observation i, which can be obtained in this way for each of the n observations. The sum of the error OOB and the average importance of all OOB trees determine the total and the relative importance of selected variables.

2.2.3 Importance of variables

In Random Forest, a ranking of the contribution of each variable is determined to predict the output variable [23], establishing a relative importance of them. This value is calculated using two different measures. The first measure the MDA (Mean Decrease Accuracy), which is based on the contribution of each variable to the prediction error (MSE for regression) and the percentage of misclassifications (for classification). The second measure of importance, MDG (Mean Decrease Gini) from the Gini index, which is the criterion used to select each partition in the construction of the trees. A decrease of the error attributed to a variable, its contribution will be lower for all trees.

For each tree t, we consider the error associated with a sample as OOBt, errOOBt denoted as the error of a singletree t OOBt sample. Randomly permute the values of Xj in OOBt to get a permuted sample and calculate their errOOBjt OOBtj as predictor error on the permuted sample t. Thus express the importance of variables (VI) as:

$$VI(X^j) = \frac{1}{ntree}\sum_t(err\widetilde{OOB}_{tJ} - errOOB_t) \quad (1)$$

A large value of *VI* indicates the importance of the predictor. By similarity, in the context of classification Bagging, we add the contribution of the Gini index and the decrease in each partition on a given as average for all predictor trees.

The Gini index (see formula 2) measures the classification error committed in node *t* yet being this leaf, the class assigned randomly instance, following the distribution of elements in each class in *t*. The Gini index for a node *t* can be calculated as:

$$i(t) = \sum_{i\neq j}^{c} P_i P_J = 1 - \sum_{j}^{c} P_j^2 \quad (2)$$

Where *c* is the number of classes and P_i is the estimated probability of class *i* for instances that reach the node. Therefore, the Gini index and information gain are measures based on the impurity of each node.

3 Experimentation

In any model of computational intelligence it is important to establish and determine the parameters that will enable us to adjust this model. In RF it must be determined the correct number of trees, and establish how many predictors are used in the construction of each tree node. A reasonable strategy for accomplishing this is to set different values and evaluate the prediction error condition.

3.1 Setting the model

The model behaviour is influenced by two parameters: the number of trees needed and the number of partitions to be made (splits). In this section the influence and the optimal values for these parameters are analysed.

3.1.1 Number of trees

Among the main parameters that can be set in RF, using the R package [25] we found the *ntree*, which sets the number of trees used in the model. We note that as the size of the tree grows in terms of number of nodes, their accuracy on the

training set improves until it stabilizes. For the three dataset can be estimated that the resulting error OOB is quite low for all cases. With a value of 300 trees *ntree*, the error remains stable. However, for a small number of trees can be seen that this leads to an overfitting model on the training data (Figure 3).

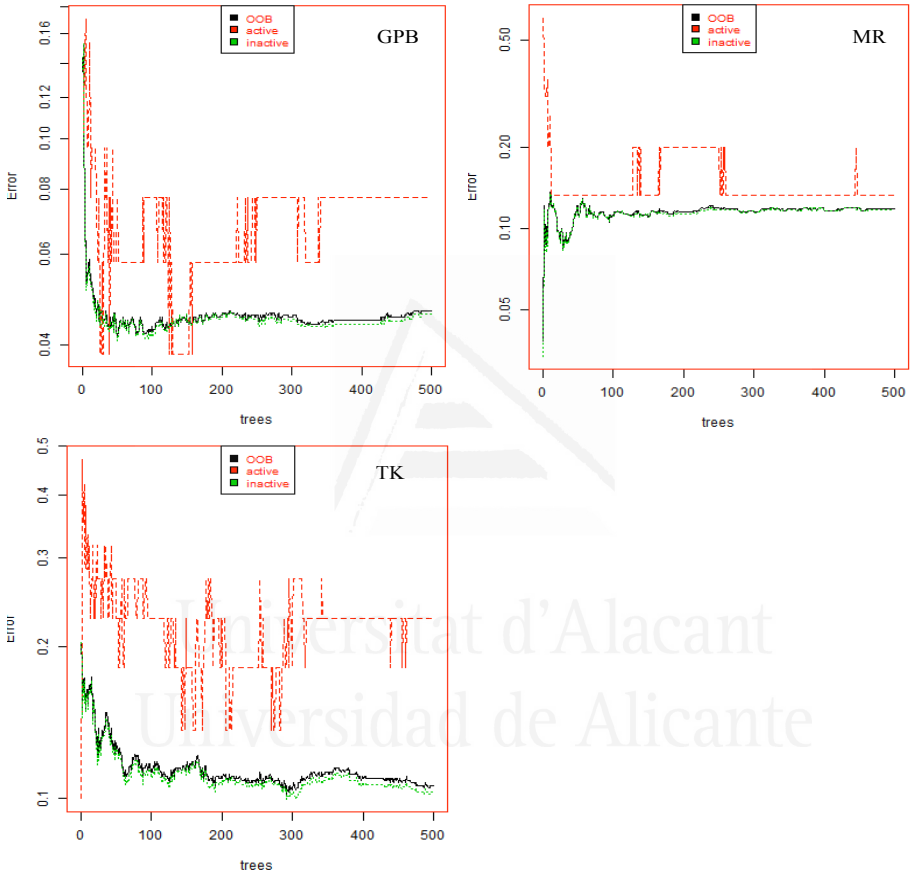


Fig. 3. OOB error (black line), active misclassify (red line) and inactive misclassify (green line) vs. number of trees for the dataset GBP, MR and TK.

3.1.2 Number of splits

The other main parameter that can set in the RF package is *mtry* R, which represents the number of input variables to be used in each node.

To construct each forest tree in RF, whenever a tree is divided is considered a random sample of "*m*" are chosen predictors of the complete set of "*p*" input predictors (molecular descriptors). These splits can choose only *m* predictors, usually the square root of the number of input predictors for classification and a third part of these are used for regression.

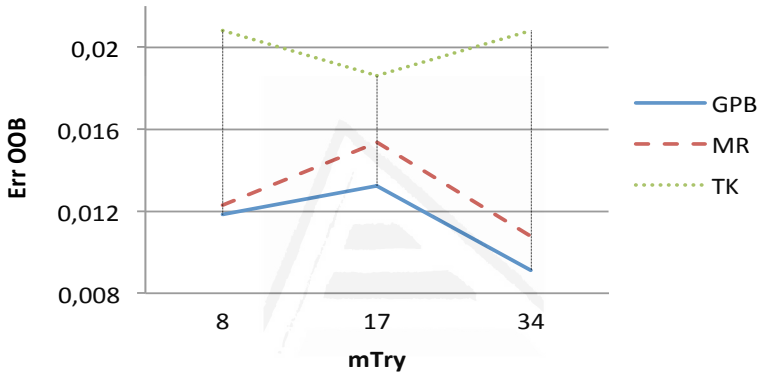


Fig. 4. Relationship between OOB error and *mtry*.

As we can see in the graph that estimates the minimum OOB error, the lowest error it occurs when *mtry* take values between 17 and 34 for GPB and MR datasets. A minimum value close to 0.013 is reached in the case of MR, we can set the value to *mtry* as the square root of the number of predictors, by default (Figure 4). We may also use a previous resampling featuring RF packet (TuneRF), estimating an optimal value for minimizing the OOB mtry error for each dataset.

3.2 Automatic selection of features

The relative importance of the variables within each dataset determines the automatic selection of molecular descriptors used. In our experiments we can observe the input and differentiating these from the dataset.

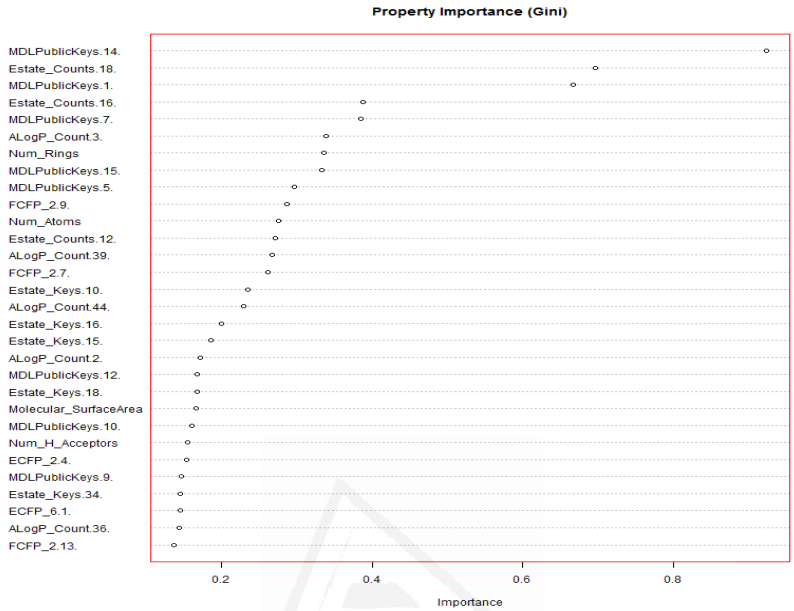


Fig. 5. Relative importance of the predictors for the dataset MR.

For different molecular dataset and for each descriptor, we can observe the importance of the contribution to predict the model and determine the sensitivity of these with respect to the prediction of the final activity (Figure 5 and Table 4).

Table 4. Top 10 molecular descriptors for dataset (ordered by relative importance).

Order	TK	MR	GPB
1	FCFP_2.12	MDLPublicKeys.14	Estate_Keys.13
2	ALogP_Count.48	Estate_Counts.18	ALogP_Count.56
3	MDLPublicKeys.12	MDLPublicKeys.1	ALogP_Count.8
4	Estate_Keys.34	Estate_Counts.16	Estate_Keys.34.
5	ECFP_4.5	MDLPublicKeys.7	Estate_Counts.34
6	ALogP_Count.56	ALogP_Count.3	Estate_Counts.13
7	Estate_Keys.9	Num_Rings	MDLPublicKeys.1
8	FCFP_4.12	MDLPublicKeys.15	ECFP_4.12
9	ALogP_Count.72	MDLPublicKeys.5	MDLPublicKeys.15
10	ECFP_6.1	FCFP_2.9	Num_H_Donors

The selection of descriptors was performed according to the dataset, using Random Forest for the selection of variables, and then to the classification of the

previous selection, AUC determine the goodness of the fitting for the prediction of the activity (Figure 6).

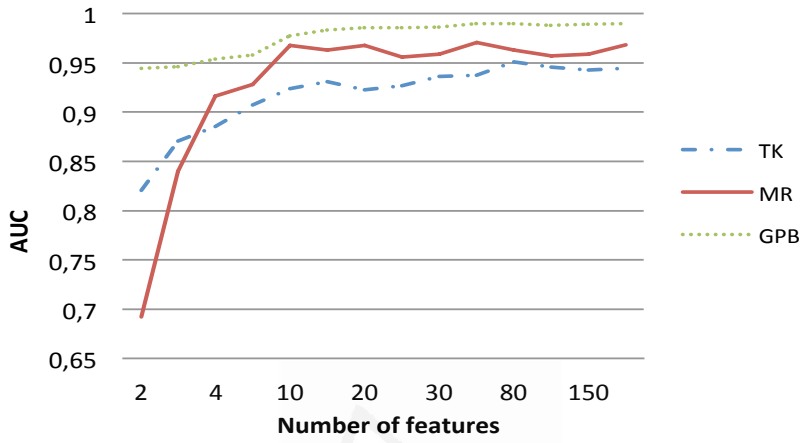


Fig. 6. AUC vs. Number of features (ordered by relative importance).

We observe that the number of significant variables (relative importance) predicting the final activity varies with the dataset used. In the worst case, with AUC of 80 characteristics and the best, with only 20 features obtains the maximum value of AUC for a dataset (GBP).

4 Results and discussion

Random Forest selects automatically the molecular descriptors that allow improving the goodness of the fitting process, considering that this selection of features depends on the dataset used. We developed a set of experiments to test the validity of our initial hypothesis with an automatic selection of molecular descriptors, compared to the manual method (ad hoc).

From the data obtained using this technique for selection of variables, we can retrain the model databases of known active or inactive compounds (Tables 5-7). This information will be used later to improve predictions and contribute to improved performance and acceleration in the discovery of new drugs using virtual screening techniques.

Faced with the manual selection of molecular descriptors using only a small set of representative chemical properties, the automatic selection uses a larger set of these, as their field of exploration and comparison is much larger, allowing a better fit and goodness in predicting the activity.

Table 5. Results of the AUC values or descriptors selection and classification using Random Forest DUD based on data for GPB, TK and MR.

	GPB	TK	MR
GPB	0.9882998	0.9506173	0.9647799
TK	0.9880257	0.9458729	0.9558176
MR	0.9898949	0.9445465	0.9571279

We have presented aspects of the problem of automatic feature selection. This paper covers the challenges of feature selection through computational intelligence methods. In addition, we proposed a solution and an alternative to traditional manual selection of features (ad hoc), which requires a very precise knowledge of the scope of the domain, and sometimes the involvement of multiple disciplines or experts in the problem to predict.

Table 6. Top of values obtained for the AUC of the ROC curves for the DUD data sets TK, MR, GPB and BINDSURF processed by NNET and SVM.

TK		MR		GPB	
NNET_EE246	0.94	NNET_EstCt	0.87	NNET_EAE246	0.96
SVM_AE246	0.95	SVM_EstKy	0.98	SVM_AlCnt	0.98
BINDSURF	0.70	BINDSURF	0.70	BINDSURF	0.68

The use of Random Forest can isolate the selection of molecular descriptors of the dataset used and do it automatically, ensuring the best possible prediction of activity in an automated way. The use of this method for the classification of the final prediction of the final activity improves the goodness of fit manual sorting.

Table 7. Top of values obtained for the AUC of the ROC curves for the DUD data sets TK, MK, GPB and BINDSURF processed by NNET, SVM using a manual selection of descriptor against automatic selection processed by RF.

Descriptor	TK		MR		GPB	
Ad Hoc	NNET_EE246	0.94	NNET_EstCt	0.87	NNET_EAE246	0.96
Ad Hoc	SVM_AE246	0.95	SVM_EstKy	0.98	SVM_AlCnt	0.98
	BINDSURF	0.70	BINDSURF	0.70	BINDSURF	0.68
Auto	C_RF_RF	0.95	C_RF_RF	0.98	C_RF_RF	0.99

5 Conclusions

In this work we have demonstrated the capacity of the automatic selection of characteristics, molecular descriptors, using the technique of Random Forest, facing the manual selection of descriptors (ad hoc). Improving the prediction of the activity is explained by improving the goodness of the fitting and its value is expressed by the AUC of the ROC curves.

Faced with the manual selection of molecular descriptors using only a small set of representative chemical properties, the automatic selection uses a larger set of these for his field of exploration and comparison time is much higher. This large number of variables makes the results and conclusions using these features to exceed the understanding of the domain to which they belong, and precludes the search for relationships between molecular descriptors that have led to the selection of features, outside the strict computational fieldis still very difficult to interpret the selection.

However, it should be mentioned that the computational intelligence approaches could be used only when there is dataset available of active and inactive compounds. Given the improvements shown in the results, it is concluded that this methodology can be used to improve the drug design and discovery, reusing and therefore helping considerably in biomedical research.

Acknowledgements

This work was partially funded by the projects: NILS Mobility Project 012-ABEL-CM-2014A and Fundacion Seneca 18946/JLI/13.

References

[1] H. Pérez-Sánchez, G. Cano, and J. García-Rodríguez, “Improving drug discovery using hybrid softcomputing methods,” Appl. Soft Comput., vol. 20, pp. 119–126, 2014.

[2] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.

[3] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, “Feature extraction,” Found. Appl., 2006.

[4] N. Huang, B. K. Shoichet, and J. J. Irwin, “Benchmarking Sets for Molecular Docking,” J. Med. Chem., vol. 49, no. 23, pp. 6789–6801, 2006.

[5] D.-S. Cao, Q.-S. Xu, Q.-N. Hu, and Y.-Z. Liang, “ChemoPy: freely available python package for computational biology and chemoinformatics,” Bioinforma. , vol. 29 , no. 8 , pp. 1092–1094, Apr. 2013.

[6] G. James, D. Witten, T. Hastie, and R. Tibshirani, An introduction to statistical learning. Springer, 2013.

[7] K. Fukunaga, Introduction to statistical pattern recognition. Academic press, 1990.

[8] S.-W. Lee, Advances in handwriting recognition, vol. 34. World Scientific, 1999.

[9] X. Huang, A. Acero, H.-W. Hon, and R. Foreword By-Reddy, Spoken language processing: A guide to theory, algorithm, and system development. Prentice Hall PTR, 2001.

[10] T. Y. Young, Handbook of pattern recognition and image processing (vol. 2): computer vision. Academic Press, Inc., 1994.

[11] A. W.-C. Liew, H. Yan, and M. Yang, "Pattern recognition techniques for the emerging field of bioinformatics: A review," Pattern Recognit., vol. 38, no. 11, pp. 2055–2073, 2005.

[12] E. S. Berner, Clinical Decision Support Systems. Springer, 2007.

[13] Y.-Y. Nguwi and A. Z. Kouzani, "Automatic road sign recognition using neural networks," in Neural Networks, 2006. IJCNN'06. International Joint Conference on, 2006, pp. 3955–3962.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in Learning for Text Categorization: Papers from the 1998 workshop, 1998, vol. 62, pp. 98–105.

[15] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[16] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[17] R. Harb, X. Yan, E. Radwan, and X. Su, "Exploring precrash maneuvers using classification trees and random forests," Accid. Anal. Prev., vol. 41, no. 1, pp. 98–107, 2009.

[18] D. Longjun, L. Xibing, X. Ming, and L. Qiyue, "Comparisons of random forest and support vector machine for predicting blasting vibration characteristic parameters," Procedia Eng., vol. 26, pp. 1772–1781, 2011.

[19] B. Larivière and D. den Poel, "Predicting customer retention and profitability by using random forests and regression forests techniques," Expert Syst. Appl., vol. 29, no. 2, pp. 472–484, 2005.

[20] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," Expert Syst. Appl., vol. 36, no. 3, pp. 5445–5449, 2009.

[21] P. Xu and F. Jelinek, "Random forests and the data sparseness problem in language modeling," Comput. Speech Lang., vol. 21, no. 1, pp. 105–152, 2007.

[22] Y.-S. Ding and T.-L. Zhang, "Using Chou's pseudo amino acid composition to predict subcellular localization of apoptosis proteins: an approach with immune genetic algorithm-based ensemble classifier," Pattern Recognit. Lett., vol. 29, no. 13, pp. 1887–1892, 2008.

[23] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, The elements of statistical learning, vol. 2, no. 1. Springer, 2009.

[24] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, no. 2, pp. 123–140, 1996.

[25] R. C. Team and others, "R: A language and environment for statistical computing," 2012.

B. SUPPORT VECTOR MACHINES PREDICTION OF DRUG SOLUBILITY ON GPUS

Submitted to the International Journal of Parallel Programming.

Resumen: El panorama actual de la computación de alto rendimiento se abre a grandes oportunidades en la simulación de los sistemas biológicos relevantes y aplicaciones de la bioinformática, biología computacional y química computacional.

El uso de grandes bases de datos no solo se aumentan las posibilidades de encontrar mejores candidatos, sino que también lo hace el tiempo de cálculo necesario en relación al número datos y la exactitud del método de CV.

En este trabajo se discuten las ventajas de utilizar las arquitecturas masivamente paralelas para la predicción de la solubilidad de compuestos, usando métodos de inteligencia computacional como las maquinas de soporte vectorial (*SVM*). Las SVMs son entrenadas con una base de datos de compuestos solubles e insolubles, esta información será explotada posteriormente para mejorar la predicción mediante CV.

Se ha demostrado empíricamente que las arquitecturas GPU se adaptan perfectamente a la aceleración de los métodos de inteligencia computacional. En el caso de las *SVM*, podemos obtener una aceleración de hasta 15 veces en relación a su equivalente de la versión secuencial.

Support Vector Machines Prediction of drug solubility on GPUs

Gaspar Cano[1], José García-Rodríguez[1] and Horacio Pérez-Sánchez[2]

[1]Dept. of Computing Technology, University of Alicante, PO.Box. 99.
E03080. Alicante, Spain {jgarcia,gcano}@dtic.ua.es

[2] Computer Science Department, Catholic University of Murcia (UCAM)
E30107. Murcia, Spain {hperez}@ucam.edu

Abstract. The landscape in the high performance computing arena opens up great opportunities in the simulation of relevant biological systems and for applications in Bioinformatics, Computational Biology and Computational Chemistry. Larger databases increase the chances of generating hits or leads, but the computational time needed for the calculations increases not only with the size of the database but also with the accuracy of the Virtual Screning (VS) methods and the model.

In this work we discuss the benefits of using massively parallel architectures for the optimization of prediction of compound solubility using computational intelligence methods such as Support Vector Machines (SVM) methods. SVMs are trained with a database of known soluble and insoluble compounds, and this information is being exploited afterwards to improve VS prediction.

We empirically demonstrate that GPUs are well-suited architecture for the acceleration of Computational Intelligence methods as SVM, obtaining up to a 15 times sustained speedup compared to its sequential counterpart version.

Keywords: SVM, GPU, CUDA, Bioinformatics, Computational Biology.

1 Introduction

The discovery of new drugs is a complicated process that can enormously profit, in the first stages, from the use of Virtual Screening (VS) methods. The limitations of VS predictions are directly related to a lack of computational resources, a major bottleneck that prevents the application from detailed, high-accuracy models to VS. However, the emergent massively parallel architectures, Graphics Processing Units (GPU), are continuously demonstrating great

performances in a wide variety of applications and, particularly, in such simulation methods [1].
The newest generations of GPUs are massively parallel processors that can support several thousand concurrent threads. Current NVIDIA [2] GPUs contain scalar processing elements per chip and are programmed using C language extensions called CUDA (Compute Unified Device Architecture) [3] . On GPUs, speedup increase reaches 100 times [4], while achieve a 200 times acceleration [5]. In NVIDIA Kepler architecture some GPUs models reached a peak performance above 3.52 TFLOPS [6].

In this paper, we focus on the optimization of the calculation of the solubility prediction using computational intelligence methods as support vector machines (SVM). SVMs are trained with a database of known soluble and insoluble compounds, and this information is being exploited afterwards to improve VS prediction.

The rest of the paper is organized as follows. Section 2 introduces the GPU architecture and CUDA programming model from NVIDIA. Section 3 shows intelligence computational methods as SVM, explains dataset and the molecular descriptors are the input for the prediction. Section 4 presents the experimentation SVM CPU vs. GPU. The performance evaluation is discussed in the Section 4. Finally, Section 5 ends with some conclusions and ideas for future work.

2 Methodology

This section describes the methods we used to improve the prediction of solubility. A computational intelligence technique such as SVM is used for the prediction of solubility and this method is trained with different datasets, and shows the GPU architecture used to accelerate the process, the dataset and molecular descriptor used.

2.1 GPU architecture and CUDA overview

As we mentioned previously, the process of apply computional intelligence methods to predict the solubility in large databases of elements is time consuming. To accelerate this process we propose the use of high performance hardware. GPUs are devices composed of a large number of processing units that were initially designed to assist the CPU in graphics-related calculations but have evolved in the course of time into programmable devices capable of tremendous performance in terms of standard integer and floating-point arithmetic.

When programmed correctly, GPUs can be used as general-purpose processors. Previous code for serial processors have to be readapted to these architectures, and this is not always a straightforward process. During the last four years, an increased programming feasibility and the introduction of new programming tools and languages has enabled a broader research community to exploit GPU programming.

GPUs are highly parallel devices with hundreds or thousands of cores. Special mathematical functions widely used in scientific calculations e.g. trigonometric and square root operations, are directly implemented in the hardware, allowing a fast performing of such calculations. The theoretical peak performance varies between the different GPU models, the NVIDIA GPU architecture is based on scalable processor array which streaming processors (SPs) cores organized as streaming multiprocessors (SMs) and off-chip memory called device memory. Each SM contains SPs with on-chip shared memory, which has very low access latency (see figure 1).

The CUDA programming model allows writting parallel programs for GPUs using some extensions of the C language. A CUDA program is divided into two main parts: the program which run on the CPU (host part) and the program executed on the GPU (device part), which is called kernel. In a kernel there are two main levels of parallelism: CUDA threads, and CUDA thread blocks[7] .

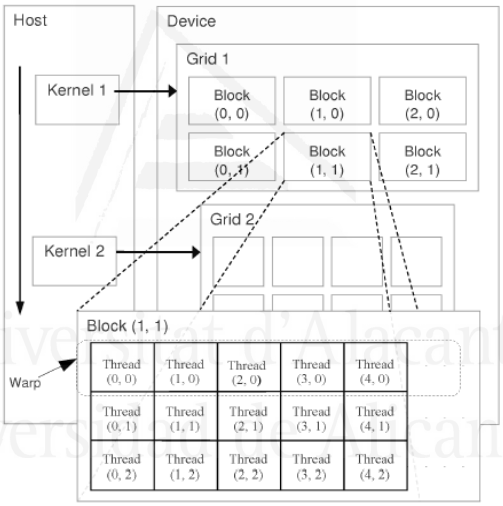


Fig. 1. Flowchart of NVIDIA GPU system.

2.2 Support Vector Machines

Support vector machines (SVM) [8] are a group of supervised learning methods that can be applied to classification or regression. They represent the decision boundary in terms of a typically small subset of all training examples, called the support vectors. In a short period of time, SVM have found numerous applications in chemistry, such as drug design (discriminating between ligands and no ligands, inhibitors and no inhibitors, etc.) [9], drug discovery [10],

quantitative structure-activity relationships (QSAR), where SVM regression is used to predict various physical, chemical, or biological properties) [11] chemometrics (optimization of chromatographic separation or compound concentration prediction from spectral data as examples), and sensors (for qualitative and quantitative prediction from sensor data), chemical engineering (fault detection and modelling of industrial processes) [12]. An excellent review of SVM applications in chemistry was published by Ivancicuc [13].

In our case, we exploit the idea that SVM produce a particular hyperplane in feature space that separates soluble or insoluble compounds, called the maximum margin hyperplane (see Figure 3). Most used kernels within SVM include: linear, Polynomial, Neural (sigmoid, Tanh), Anova, Fourier, Spline, B Spline, Additive, Tensor and Gaussian Radial Basis or Exponential Radial Basis.

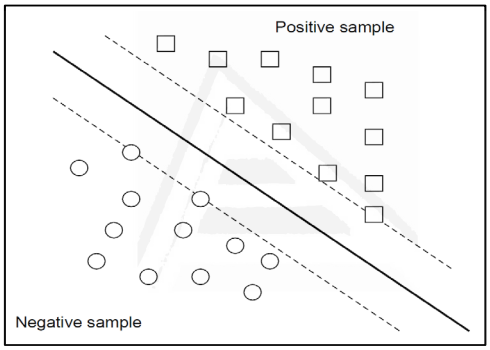


Fig. 2. Support Vector Machine margin hyperplanes.

2.3 Ligand Databases and molecular properties

We employed standard VS benchmark tests, such as NCI (Release 4 – May 2012) is large database of molecules [14]. Next, using the ChemoPy package [15] we calculated for all ligands of the sets a diverse of molecular properties derived from the set of constitutional, CPSA (charged partial surface area) and fragment/fingerprint-based descriptors, as described in Table 1. Constitutional properties depend on very simple descriptors of the molecule that can be easily calculated just counting the number of molecular elements such as atoms, types of atoms, bonds, rings, etc. These descriptors should be able to differentiate very dissimilar molecules, but might have problems for separating closely related isomers. CPSA descriptors consider finer details of molecular structure, so they might be able to separate similar molecules, but might also have difficulties for separating isomers. Lastly, fragment and fingerprint-based descriptors take into account the presence of an exact structure (not a substructure) with limited specified attachment points. In the generation of fingerprints, the program assigns

an initial code to each atom. The initial atom code is derived from the number of connections to the atom, the element type, atomic charge, and atomic mass. This corresponds to an ECFP with a neighbourhood size of zero. These atom codes are then updated in an iterative manner to reflect the codes of each atoms neighbour. In the next iteration, a hashing scheme is employed to incorporate information from each atom immediate neighbour. Each atom new code now describes a molecular structure with a neighbourhood size of one. This process is carried out for all atoms in the molecule. When the desired neighbourhood size is reached, the process is complete and the set of all features is returned as the fingerprint. For the ECFPs employed in this paper, neighbourhood sizes of two, four and six (ECFP 2, ECFP 4, ECFP 6) were used to generate the fingerprints. The resulting ECFPs can represent a much larger set of features than other fingerprints and contain a significant number of different structural units crucial for the molecular comparison, among the compounds.

CONSTITUTIONAL DESCRIPTORS	
Natom	Number of atoms
MolWe	Molecular Weight
NRing	Number of rings
NArRg	Numer of aromatic rings
NRotB	Number of rotatable bonds
NHDon	Number of H-bond donors
NHAcc	Number of H-bond acceptors
CPSA DESCRIPTORS	
Msurf	Molecular surface area
Mpola	Molecular polar surface area
Msolu	Molecular solubility
AlogP	Partition coefficient
FRAGMENT/FINGERPRINT-BASED DESCRIPTORS	
ECP2, ECP4, ECP6	Extended-connectivity fingerprints (ECFP)
EstCt	Estate counts
AlCnt	AlogP2 Estate counts
EstKy	Estate keys
MDLPK	MDL public keys

Table 1. Molecular descriptors used in this study.

In order to discriminate which is the best option for discriminating between soluble and insoluble compounds in these datasets we use fingerprint-based descriptors and avoid the use of constitutional and CPSA descriptors. This is reasonable since fingerprint descriptors consider more details about the structure of molecules, being able to efficiently discriminate with more accuracy between active compounds and their decoys.

COMBINATIONS OF CONSTITUTIONAL DESCRIPTORS	
MNBH	Molecular polar surface area (MPola)+ Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MNB	Molecular polar surface area (MPola) + Number of rotatable bonds (NRotB)
NBH	Number of rotatable bonds (NRotB) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
MoN	Molecular polar surface area (MPola) + Number of H-Bond acceptors (NHAcc)
COMBINATIONS OF FRAGMENT/FINGERPRINT-BASED DESCRIPTORS	
EAE246	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EA	Estate counts (EstCt) + AlogP2 Estate counts (AlCnt)
AE246	AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)
EE246	Estate counts (EstCt) + Extended-connectivity fingerprints (ECFP)

Table 2. Combinations of molecular descriptors used in this study.

Next, we studied which combination of properties could lead to improvements on the predictive capability of these soft computing methods; it is clear that predictive capability increases (see Table 2).

3 Experimentation SVM

In this section we outline the experimentation of SVM over R[16] packages for the computational intelligence methods both for CPU and GPU versions. We will compare the experimentation of the sequential version of SVM vs. parallel version of SVM in the prediction of solubility, with one simple molecular descriptor (ALOGP) and one complex molecular descriptor (EAE246).

3.1 Sequential version of SVM (CPU)

The package e1071 is based over the library LIBSVM [17] is currently working on some methods of efficient automatic parameter selection. It is a sequentially package that makes use of the CPU.

In R there are different implementations of SVM, in order to validate the results, we have chosen this package since it has a good performance on standardized dataset. In the Table 3, we compare the four SVM [18] implementations in terms of training time. In this comparison we only focus on

the actual training time of the SVM excluding the time needed for estimating the training error or the cross-validation error. In these implementations we used the SVM package in R. The dataset used is the standard dataset from UCI Repository of Machine Learning Databases." University of California [19].

Dataset	Kernlab	E1071	klaR	svmpath
spam	18.50	17.90	34.80	34.00
musk	1.40	1.30	4.65	13.80
Vowel	1.30	0.30	21.46	NA
DNA	22.40	23.30	116.30	NA
BreastCancer	0.47	0.36	1.32	11.55
HouseBoston	0.72	0.41	92.30	NA

Table 3. Training times for the SVM implementations on different datasets in seconds. An AMD Athlon 1400 MHz computer running Linux was used.

3.2 Parallel version of SVM (GPU)

The parallel version of SVM used is part of the package RPUDPRO [20], which is building round of a parallel version of SVMLIB. To process the parallel version of SVM, we consider that the problem of classification with SVM can be separated into two different tasks; the calculation of the kernel matrix (KM) and the core SVM task of decomposing and solving the classification model. The increasing size of the input data leads to a huge KM that cannot be calculated and stored in memory (see Algorithm 1). Therefore, the solver needs to calculate on-the-fly portions of the KM, which is a processing and memory-bandwidth intensive procedure. LIBSVM is using double precision for calculations but only the latest GPUs do support double precision. The pre-calculation is performed combining CPU and GPU to achieve maximum performance.

1: Pre-calculate on CPU the elements for each training vector.
2: Convert the training vectors array into columns wise format.
3: Allocate device memory on the GPU the training vectors array.
4: Load the training vectors array to the GPU memory.
5: FOR (each training vector) DO
- Load the training vector to the GPU.
- Perform operations with CUBLAS.
- Retrieve the dot products vector from the GPU.

END DO
6: De-allocate memory from GPU.

Algorithm 1. Parallel SVM pseudo code.

As can be seen into pseudocode of the Algorithm 1, the proposed algorithm is based on recalculating on the CPU for each training vector calculation using the CUBLAS [15] library provided from NVIDIA.

4 Experimentation

The performance of sequential and GPU implementations are evaluated in a dual-core Intel E6400 (Conroe with 2 MB L2 cache), which acts as a host machine for our NVIDIA GeForce GT430 GPU. In previous papers, we show that selection of variables is very import step for the best prediction [21].

The benchmarks are executed by varying the number of molecules from the dataset and choosing different molecular descriptors as predictor for solver our classification problem (solubility or insolubility). From the simple molecular descriptors consisting of a number of small items such as CPSA descriptors (Msurf, Mpola, Msolu, AlogP) or constitutional descriptors (Natom, MolWe, NRing, NArRg, NRotB, NHDon, NHAcc) that have only a molecular features. The calculation of fragmement/fingerprint-based descriptors, vectors with some molecular characteristics, is computationally heavy. The combinations of constitutional descriptors (MNBH, MNB, NBH, MoN) or combinations of fragment/fingerprint-based descriptors (EAE246, EA, AE246) are the mix of the previous simples descriptors, and need more process time.

Results were obtained for different number of molecules from dataset, calculating the execution time with SVM with CPU and GPU versions. As predictor (input data), we choose a computational easy molecular descriptor (ALogp, Partition coefficient) and a computational heavy molecular descriptor (EAE246, Estate counts (EstCt) + AlogP2 Estate counts (AlCnt) + Extended-connectivity fingerprints (ECFP)).

We note that, the speedup factor between GPU and CPU increases faster e when the number of molecules in dataset is higher. This is because the number of molecules will be increasing the number of thread blocks running in parallel, and then the GPU resources are fully used. However, it performs similarly compared to the GPU for the smallest benchmarks in which the GPU is not fully used, for the small molecular descriptor (ALOGP) and for a large number of molecules we have an speedup of 2x (GPU vs. CPU, see Figure 3). For complex molecular descriptors since a low number of molecules are use, the speedup to CPU vs. GPU is very significant. As shown in Figure 4 for large numbers of molecules and complex descriptor (EAE246), we have speedups of 15x (GPU vs. CPU).

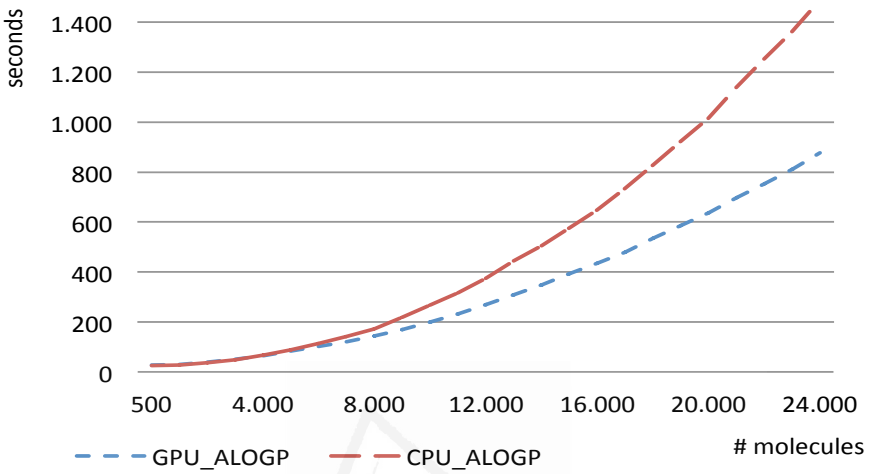


Fig.3. Speedup GPU vs. CPU for simple descriptor (ALOGP). Increasing the runtime in seconds when increase the number of molecules (speeup x2).

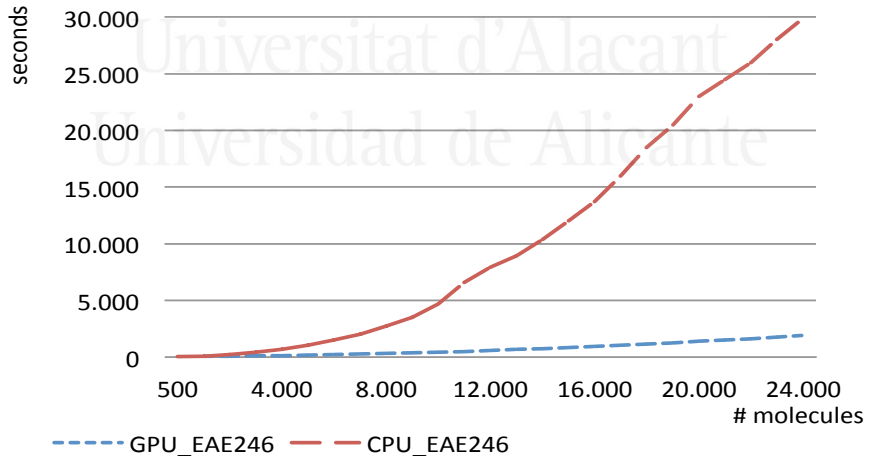


Fig.4. Speedup GPU vs. CPU for complex descriptors (EAE246). Increasing the runtime in seconds when increase the number of molecules (speedup x15).

5 Conclusions and Future Work

In this paper we have introduced the benefits of using massively parallel architectures for the optimization of prediction of solubility using computational intelligence methods as support vector machines (SVM). SVMs are trained with a database of known soluble and insoluble compounds, and this information is being exploited afterwards to improve VS prediction for different emergent parallel architectures. The results obtained for GPU are indeed promising, given the obtained speedup values up to 15, compared with the sequential version, and the progression is up.

The good results exposed open the way to use large databases for prediction of solubility, using computational intelligence methods as support vector machines (SVM). These methods fit well in the GPU architecture for a bigger numbers of molecules, or complex descriptors that are necessary to obtain good values in the prediction.

As future work we will use other computational intelligence methods, for classification, regression or selection of variables.

Acknowledgements

This work was partially funded by the projects: NILS Mobility Project 012-ABEL-CM-2014A and Fundacion Seneca 18946/JLI/13. Experiments were made possible with a generous donation of hardware from NVDIA.

References

[1] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference*, 2007, pp. 746–749.

[2] W. Nvidia, N. Generation, and C. Compute, "Whitepaper NVIDIA's Next Generation CUDA Compute Architecture:," pp. 1–22.

[3] C. Nvidia, "Compute unified device architecture programming guide," 2007.

[4] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH Computer Architecture News*, 2007, vol. 35, no. 2, pp. 13–23.

[5] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, 2004, pp. 4–10.

[6] A. Ruiz and M. Ujaldón, "Acelerando los momentos de Zernike sobre Kepler," 2014.

[7] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *Comput. J.*, vol. 53, no. 7, pp. 1045–1051, 2010.

[8] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[9] R. N. Jorissen and M. K. Gilson, "Virtual Screening of Molecular Databases Using a Support Vector Machine," *J. Chem. Inf. Model.*, vol. 45, no. 3, pp. 549–561, Apr. 2005.

[10] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen, "Active learning with support vector machines in the drug discovery process," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 2, pp. 667–673, 2003.

[11] J. M. Kriegl, T. Arnhold, B. Beck, and T. Fox, "Prediction of Human Cytochrome P450 Inhibition Using Support Vector Machines," *QSAR Comb. Sci.*, vol. 24, no. 4, pp. 491–502, 2005.

[12] D. E. Lee, J.-H. Song, S.-O. Song, and E. S. Yoon, "Weighted Support Vector Machine for Quality Estimation in the Polymerization Process," *Ind. Eng. Chem. Res.*, vol. 44, no. 7, pp. 2101–2105, Mar. 2005.

[13] O. Ivanciuc, "Applications of Support Vector Machines in Chemistry," in *Reviews in Computational Chemistry*, John Wiley & Sons, Inc., 2007, pp. 291–400.

[14] J. H. Voigt, B. Bienfait, S. Wang, and M. C. Nicklaus, "Comparison of the NCI open database with seven large chemical structural databases," *J. Chem. Inf. Comput. Sci*, vol. 41, no. 3, pp. 702–712, 2001.

[15] D.-S. Cao, Q.-S. Xu, Q.-N. Hu, and Y.-Z. Liang, "ChemoPy: freely available python package for computational biology and chemoinformatics," *Bioinforma.* , vol. 29 , no. 8 , pp. 1092–1094, Apr. 2013.

[16] R. C. Team and others, "R: A language and environment for statistical computing," 2012.

[17] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[18] K. Hornik, D. Meyer, and A. Karatzoglou, "Support vector machines in R," *J. Stat. Softw.*, vol. 15, no. 9, pp. 1–28, 2006.

[19] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science." 1998.

[20] Yau, "GPU Computing with R.' R Tutorial: An R Introduction to Statis,'" *r - tutor.com/*, 2014. .

[21] H. Pérez-Sánchez, G. Cano, and J. García-Rodríguez, "Improving drug discovery using hybrid softcomputing methods," *Appl. Soft Comput.*, vol. 20, pp. 119–126, 2014.

REFERENCIAS

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank.," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–42, Jan. 2000.

[2] R. Sanchez and A. Sali, "Large-scale protein structure modeling of the Saccharomyces cerevisiae genome," *Proc. Natl. Acad. Sci.*, vol. 95, no. 23, pp. 13597–13602, Nov. 1998.

[3] T. A. Halgren, R. B. Murphy, R. A. Friesner, H. S. Beard, L. L. Frye, W. T. Pollard, and J. L. Banks, "Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening," *J. Med. Chem.*, vol. 47, no. 7, pp. 1750–1759, 2004.

[4] O. Trott and A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *J. Comput. Chem.*, vol. 31, no. 2, pp. 455–461, 2010.

[5] T. J. Ewing, S. Makino, a G. Skillman, and I. D. Kuntz, "DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases.," *J. Comput. Aided. Mol. Des.*, vol. 15, no. 5, pp. 411–28, May 2001.

[6] I. Sánchez-Linares, H. Pérez-Sánchez, J. M. Cecilia, and J. M. García, "High-Throughput parallel blind Virtual Screening using BINDSURF," *BMC Bioinformatics*, vol. 13, no. Suppl 14, p. S13, 2012.

[7] M. Garland and D. B. Kirk, "Understanding throughput-oriented architectures," *Commun. ACM*, vol. 53, no. 11, p. 58, Nov. 2010.

[8] M. Pharr and R. Fernando, "2: Programming Techniques for High-performance Graphics and General-purpose Computation (gpu gems)." Addison-Wesley Professional, 2005.

[9] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, "Parallel Computing Experiences with CUDA," *Micro, IEEE*, vol. 28, no. 4, pp. 13–27, 2008.

[10] W. Nvidia, N. Generation, and C. Compute, "Whitepaper NVIDIA's Next Generation CUDA Compute Architecture:," pp. 1–22.

[11] A. Ruiz and M. Ujaldón, "Acelerando los momentos de Zernike sobre Kepler," 2014.

[12] H. Perez Sanchez and W. Wenzel, "Optimization methods for virtual screening on novel computational architectures.," *Curr. Comput. Aided. Drug Des.*, vol. 7, no. 1, pp. 44–52, 2011.

[13] G. Guerrero, H. Pérez-Sánchez, W. Wenzel, J. Cecilia, and J. García, "Effective Parallelization of Non-bonded Interactions Kernel for Virtual Screening on GPUs," in *5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2011) SE - 9*, vol. 93, M. Rocha, J. C. Rodríguez, F. Fdez-Riverola, and A. Valencia, Eds. Springer Berlin Heidelberg, 2011, pp. 63–69.

[14] I. Sánchez-Linares, H. Pérez-Sánchez, and J. M. García, "Accelerating Grid Kernels for Virtual Screening on Graphics Processing Units," in *Proceedings of the 14th International Conference on Parallel Computing - ParCo 2011*.

[15] G. Brannigan, D. N. LeBard, J. Hénin, R. G. Eckenhoff, and M. L. Klein, "Multiple binding sites for the general anesthetic isoflurane identified in the nicotinic acetylcholine receptor transmembrane domain.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 32, pp. 14122–7, Aug. 2010.

[16] H. Esmaeilzadeh, E. Blem, R. St Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, 2011, pp. 365–376.

[17] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference*, 2007, pp. 746–749.

[18] G. Klebe, "Virtual ligand screening: strategies, perspectives and limitations," *Drug Discov. Today*, vol. 11, no. 13, pp. 580–594, 2006.

[19] M. Bhagwat and L. Young, "Bioinformatics. Charlie Hodgman, Andrew French and David Westhead Instant," *Brief. Bioinform.*, vol. 12, no. 1, pp. 78–79, 2011.

[20] T. Schwede, J. Kopp, N. Guex, and M. C. Peitsch, "SWISS-MODEL: an automated protein homology-modeling server," *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3381–3385, 2003.

[21] J. Harenberg and M. Wehling, "Current and future prospects for anticoagulant therapy: inhibitors of factor Xa and factor IIa," in *Seminars in thrombosis and hemostasis*, 2008, vol. 34, no. 01, pp. 39–57.

[22] W. L. Jorgensen, "The many roles of computation in drug discovery," *Science (80-.).*, vol. 303, no. 5665, pp. 1813–1818, 2004.

[23] A. I. De Agostini, S. C. Watkins, H. S. Slayter, H. Youssoufian, and R. D. Rosenberg, "Localization of anticoagulantly active heparan sulfate proteoglycans in vascular endothelium: antithrombin binding on cultured endothelial cells and perfused rat aorta.," *J. Cell Biol.*, vol. 111, no. 3, pp. 1293–1304, 1990.

[24] J. S. Paikin, J. W. Eikelboom, J. A. Cairns, and J. Hirsh, "New antithrombotic agents—insights from clinical trials," *Nat. Rev. Cardiol.*, vol. 7, no. 9, pp. 498–509, 2010.

[25] J. Michel, N. Foloppe, and J. W. Essex, "Rigorous Free Energy Calculations in Structure-Based Drug Design," *Mol. Inform.*, vol. 29, no. 8–9, pp. 570–578, 2010.

[26] G. T. Gunnarsson and U. R. Desai, "Hydropathic interaction analyses of small organic activators binding to antithrombin," *Bioorg. Med. Chem.*, vol. 12, no. 3, pp. 633–640, 2004.

[27] L. Yao, J. A. Evans, and A. Rzhetsky, "Novel opportunities for computational biology and sociology in drug discovery: Corrected paper," *Trends Biotechnol.*, vol. 28, no. 4, pp. 161–170, 2010.

[28] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[29] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1990.

[30] S.-W. Lee, *Advances in handwriting recognition*, vol. 34. World Scientific, 1999.

[31] X. Huang, A. Acero, H.-W. Hon, and R. Foreword By-Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall PTR, 2001.

[32] T. Y. Young, *Handbook of pattern recognition and image processing (vol. 2): computer vision*. Academic Press, Inc., 1994.

[33] A. W.-C. Liew, H. Yan, and M. Yang, "Pattern recognition techniques for the emerging field of bioinformatics: A review," *Pattern Recognit.*, vol. 38, no. 11, pp. 2055–2073, 2005.

[34] E. S. Berner, *Clinical Decision Support Systems*. Springer, 2007.

[35] Y.-Y. Nguwi and A. Z. Kouzani, "Automatic road sign recognition using neural networks," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, 2006, pp. 3955–3962.

[36] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 workshop*, 1998, vol. 62, pp. 98–105.

[37] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 1, pp. 4–37, 2000.

[38] C. Apté and S. Weiss, "Data mining with decision trees and decision rules," *Futur. Gener. Comput. Syst.*, vol. 13, no. 2, pp. 197–210, 1997.

[39] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[40] A. J. Smola and B. Schölkopf, "On a kernel-based method for pattern recognition, regression, approximation, and operator inversion," *Algorithmica*, vol. 22, no. 1–2, pp. 211–231, 1998.

[41] J. W. Gardner, "Detection of vapours and odours from a multisensor array using pattern recognition Part 1. Principal component and cluster analysis," *Sensors Actuators B Chem.*, vol. 4, no. 1, pp. 109–115, 1991.

[42] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Inf. Theory, IEEE Trans.*, vol. 21, no. 1, pp. 32–40, 1975.

[43] V. Vapnik and L. Bottou, "Local algorithms for pattern recognition and dependencies estimation," *Neural Comput.*, vol. 5, no. 6, pp. 893–909, 1993.

[44] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 8, pp. 747–757, 2000.

[45] H. A. Simon, "Why should machines learn?," in *Machine learning*, Springer, 1983, pp. 25–37.

[46] R. Forsyth, *Machine learning: Principles and techniques*. Chapman & Hall, Ltd., 1988.

[47] S. Weiss and C. Kulikowski, "Computer systems that learn," 1991.

[48] P. Langley, *Elements of machine learning*. Morgan Kaufmann, 1996.

[49] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Advanced Lectures on Machine Learning*, Springer, 2004, pp. 169–207.

[50] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013.

[51] M. Minsky, S. Papert, and M. I. Perceptrons, "Press," *Cambridge, Ma*, pp. 105–110, 1969.

[52] M. H. Hassoun, *Fundamentals of artificial neural networks*. MIT press, 1995.

[53] C.-T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[54] G. Chen and X. Dong, *Methodologies, Perspectives and Applications*. World Scientific, 1998.

[55] L. V Fausett, *Fundamentals of neural networks*. Prentice-Hall, 1994.

[56] S. Y. Kung, *Digital neural networks*. Prentice-Hall, Inc., 1993.

[57] C. M. Bishop and others, “Neural networks for pattern recognition,” 1995.

[58] N. Bodor, A. Harget, and M. J. Huang, “Neural network studies. 1. Estimation of the aqueous solubility of organic compounds,” *J. Am. Chem. Soc.*, vol. 113, no. 25, pp. 9480–9483, 1991.

[59] C. Hansch, A. Leo, D. Hoekman, and S. R. Heller, *Exploring Qsar*. American Chemical Society Washington, DC, 1995.

[60] G. Schneider, P. Wrede, and others, “Artificial neural networks for computer-based molecular design,” *Prog. Biophys. Mol. Biol.*, vol. 70, no. 3, pp. 175–222, 1998.

[61] K. L. Peterson, “Artificial Neural Networks and Their use in Chemistry,” in *Reviews in Computational Chemistry*, John Wiley & Sons, Inc., 2007, pp. 53–140.

[62] W. L. Jorgensen and E. M. Duffy, “Prediction of drug solubility from structure,” *Adv. Drug Deliv. Rev.*, vol. 54, no. 3, pp. 355–366, 2002.

[63] J. Taskinen and J. Yliruusi, “Prediction of physicochemical properties based on neural network modelling.,” *Adv. Drug Deliv. Rev.*, vol. 55, no. 9, pp. 1163–83, Sep. 2003.

[64] M. Weisel, J. M. Kriegl, and G. Schneider, “Architectural Repertoire of Ligand-Binding Pockets on Protein Surfaces,” *ChemBioChem*, vol. 11, no. 4, pp. 556–563, 2010.

[65] N. R. Pal and R. Panja, “Finding short structural motifs for re-construction of proteins 3D structure,” *Appl. Soft Comput.*, vol. 13, no. 2, pp. 1214–1221, 2013.

[66] J. D. Durrant and J. A. McCammon, “NNScore: a neural-network-based scoring function for the characterization of protein-ligand complexes.,” *J. Chem. Inf. Model.*, vol. 50, no. 10, pp. 1865–71, Oct. 2010.

[67] J. D. Durrant and J. A. McCammon, “BINANA: a novel algorithm for ligand-binding characterization.,” *J. Mol. Graph. Model.*, vol. 29, no. 6, pp. 888–93, Apr. 2011.

[68] I. V. R. Reyes, I. V Fedyushkina, V. S. Skvortsov, and D. A. Filimonov, “Prediction of progesterone receptor inhibition by high-performance neural network algorithm,” . *J. Math. Model. Methods Appl. Sci.,*, 2013.

[69] G. Bortolan, R. Degani, and J. L. Willems, “ECG classification with neural networks and cluster analysis,” in *Computers in Cardiology 1991, Proceedings.*, 1991, pp. 177–180.

[70] R. Coggins, M. Jabri, B. Flower, and S. Pickard, “A low-power network for on-line diagnosis of heart patients,” *IEEE Micro*, vol. 15, no. 3, pp. 18–25, 1995.

[71] J. R. Hilera González and V. J. Martínez Hernando, “Redes neuronales artificiales: fundamentos modelos y aplicaciones,” *Madrid Editor. Alfaomega Ra-Ma*, 2000.

[72] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and others, “Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,” *Nat. Med.*, vol. 7, no. 6, pp. 673–679, 2001.

[73] R. N. Sánchez and A. J. Velásquez, “Diferenciaci{ó}n entre electrocardiogramas normales y arr{í}tmicos usando an{á}lisis en frecuencia,” *Rev. Ciencias la Salud*, vol. 2, no. 2, 2010.

[74] M. Traeger, A. Eberhart, G. Geldner, A. M. Morin, C. Putzke, H. Wulf, and L. H. Eberhart, “[Prediction of postoperative nausea and vomiting using an artificial neural network],” *Anaesthesist*, vol. 52, no. 12, pp. 1132–1138, 2003.

[75] G. J. Gibson, S. Siu, and C. F. N. Cowan, “The application of nonlinear structures to the reconstruction of binary

signals," *Signal Process. IEEE Trans.*, vol. 39, no. 8, pp. 1877–1884, 1991.

[76] G. P. Eatwell, "Noise reduction filter." Google Patents, 1998.

[77] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 20, no. 1, pp. 23–38, 1998.

[78] D. J. Burr, "Experiments on neural net recognition of spoken and written text," *Acoust. Speech Signal Process. IEEE Trans.*, vol. 36, no. 7, pp. 1162–1168, 1988.

[79] N. Suga, "Biosonar and neural computation in bats," *Sci. Am.*, vol. 262, no. 6, pp. 60–68, 1990.

[80] N. Kohzadi, M. S. Boyd, B. Kermanshahi, and I. Kaastra, "A comparison of artificial neural network and time series models for forecasting commodity prices," *Neurocomputing*, vol. 10, no. 2, pp. 169–181, 1996.

[81] A. D. Back and A. S. Weigend, "A first application of independent component analysis to extracting structure from stock returns," *Int. J. Neural Syst.*, vol. 8, no. 04, pp. 473–484, 1997.

[82] K. P. Macpherson, A. J. Conway, and J. C. Brown, "Prediction of solar and geomagnetic activity data using neural networks," *J. Geophys. Res. Sp. Phys.*, vol. 100, no. A11, pp. 21735–21744, 1995.

[83] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)--a review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, no. 14–15, pp. 2627–2636, 1998.

[84] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[85] J. M. Kriegl, T. Arnhold, B. Beck, and T. Fox, "Prediction of Human Cytochrome P450 Inhibition Using Support Vector

Machines," *QSAR Comb. Sci.*, vol. 24, no. 4, pp. 491–502, 2005.

[86] D. E. Lee, J.-H. Song, S.-O. Song, and E. S. Yoon, "Weighted Support Vector Machine for Quality Estimation in the Polymerization Process," *Ind. Eng. Chem. Res.*, vol. 44, no. 7, pp. 2101–2105, Mar. 2005.

[87] O. Ivanciuc, "Applications of Support Vector Machines in Chemistry," in *Reviews in Computational Chemistry*, John Wiley & Sons, Inc., 2007, pp. 291–400.

[88] Y. B. Dibike, S. Velickov, D. Solomatine, and M. B. Abbott, "Model induction with support vector machines: introduction and applications," *J. Comput. Civ. Eng.*, vol. 15, no. 3, pp. 208–216, 2001.

[89] Y. Zhao, B. Li, X. Li, W. Liu, and S. Ren, "Customer churn prediction using improved one-class support vector machine," in *Advanced data mining and applications*, Springer, 2005, pp. 300–306.

[90] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *Comput. Intell. Mag. IEEE*, vol. 4, no. 2, pp. 24–38, 2009.

[91] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, 2005.

[92] K. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[93] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Comput. Speech Lang.*, vol. 20, no. 2, pp. 210–229, 2006.

[94] S. E. Schwarm and M. Ostendorf, "Reading level assessment using support vector machines and statistical

language models," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 523–530.

[95] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002.

[96] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[97] K. K. Kandaswamy, K.-C. Chou, T. Martinetz, S. Möller, P. N. Suganthan, S. Sridharan, and G. Pugalenthi, "AFP-Pred: A random forest approach for predicting antifreeze proteins from sequence-derived properties," *J. Theor. Biol.*, vol. 270, no. 1, pp. 56–62, 2011.

[98] P. Zhang, T. D. Bui, and C. Y. Suen, "A novel cascade ensemble classifier system with a high recognition performance on handwritten digits," *Pattern Recognit.*, vol. 40, no. 12, pp. 3415–3429, 2007.

[99] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 130–136.

[100] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, 2005, vol. 1, pp. 947–954.

[101] Y. Zhang, W. Lee, and Y.-A. Huang, "Intrusion detection techniques for mobile wireless networks," *Wirel. Networks*, vol. 9, no. 5, pp. 545–556, 2003.

[102] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, 2002, vol. 2, pp. 1702–1707.

[103] C.-L. Huang, M.-C. Chen, and C.-J. Wang, "Credit scoring with a data mining approach based on support vector machines," *Expert Syst. Appl.*, vol. 33, no. 4, pp. 847–856, 2007.

[104] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*, 1999.

[105] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[106] R. Harb, X. Yan, E. Radwan, and X. Su, "Exploring precrash maneuvers using classification trees and random forests," *Accid. Anal. Prev.*, vol. 41, no. 1, pp. 98–107, 2009.

[107] D. Longjun, L. Xibing, X. Ming, and L. Qiyue, "Comparisons of random forest and support vector machine for predicting blasting vibration characteristic parameters," *Procedia Eng.*, vol. 26, pp. 1772–1781, 2011.

[108] B. Larivière and D. den Poel, "Predicting customer retention and profitability by using random forests and regression forests techniques," *Expert Syst. Appl.*, vol. 29, no. 2, pp. 472–484, 2005.

[109] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5445–5449, 2009.

[110] P. Xu and F. Jelinek, "Random forests and the data sparseness problem in language modeling," *Comput. Speech Lang.*, vol. 21, no. 1, pp. 105–152, 2007.

[111] Y.-S. Ding and T.-L. Zhang, "Using Chou's pseudo amino acid composition to predict subcellular localization of apoptosis proteins: an approach with immune genetic algorithm-based ensemble classifier," *Pattern Recognit. Lett.*, vol. 29, no. 13, pp. 1887–1892, 2008.

[112] M. Pardo and G. Sberveglieri, "Random forests and nearest shrunken centroids for the classification of sensor array data," *Sensors Actuators B Chem.*, vol. 131, no. 1, pp. 93–99, 2008.

[113] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemom. Intell. Lab. Syst.*, vol. 83, no. 2, pp. 83–90, 2006.

[114] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003.

[115] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, 2011.

[116] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, 2006, p. 8–pp.

[117] R. Grimm, T. Behrens, M. Märker, and H. Elsenbeer, "Soil organic carbon concentrations and stocks on Barro Colorado Island—digital soil mapping using Random Forests analysis," *Geoderma*, vol. 146, no. 1, pp. 102–113, 2008.

[118] C. Nvidia, "Compute unified device architecture programming guide," 2007.

[119] D. B. Kokh and W. Wenzel, "Flexible side chain models improve enrichment rates in in silico screening," *J. Med. Chem.*, vol. 51, no. 19, pp. 5919–5931, 2008.

[120] I. Sánchez-Linares, H. Pérez-Sánchez, J. M. Cecilia, and J. M. García, "High-Throughput parallel blind Virtual Screening using BINDSURF.," *BMC Bioinformatics*, vol. 13 Suppl 1, no. Suppl 14, p. S13, Jan. 2012.

[121] P. Wolohan and D. E. Reichert, "Use of binding energy in comparative molecular field analysis of isoform selective estrogen receptor ligands," *J. Mol. Graph. Model.*, vol. 23, no. 1, pp. 23–38, 2004.

[122] C. J. Cramer, *Essentials of computational chemistry: theories and models*. John Wiley & Sons, 2013.

[123] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *J. Am. Chem. Soc.*, vol. 117, no. 19, pp. 5179–5197, 1995.

[124] B. Waszkowycz, D. E. Clark, and E. Gancia, "Outstanding challenges in protein--ligand docking and structure-based virtual screening," *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, vol. 1, no. 2, pp. 229–259, 2011.

[125] N. Huang, B. K. Shoichet, and J. J. Irwin, "Benchmarking Sets for Molecular Docking," *J. Med. Chem.*, vol. 49, no. 23, pp. 6789–6801, 2006.

[126] D.-S. Cao, Q.-S. Xu, Q.-N. Hu, and Y.-Z. Liang, "ChemoPy: freely available python package for computational biology and chemoinformatics," *Bioinforma.* , vol. 29 , no. 8 , pp. 1092–1094, Apr. 2013.

[127] H. Perez-Sanchez and W. Wenzel, "Optimization Methods for Virtual Screening on Novel Computational Architectures," *Curr. Comput. - Aided Drug Des.*, vol. 7, no. 1, pp. 44–52, 2011.

[128] S.-Y. Huang, S. Z. Grinter, and X. Zou, "Scoring functions and their evaluation methods for protein--ligand docking: recent advances and future directions," *Phys. Chem. Chem. Phys.*, vol. 12, no. 40, pp. 12899–12908, 2010.

[129] E. Yuriev and P. A. Ramsland, “Latest developments in molecular docking: 2010--2011 in review,” *J. Mol. Recognit.*, vol. 26, no. 5, pp. 215–239, 2013.

[130] E. Evans and K. Ritchie, “Dynamic strength of molecular adhesion bonds,” *Biophys. J.*, vol. 72, no. 4, pp. 1541–1555, 1997.

[131] V. Rezaei, H. Pezeshk, and H. Pérez-Sa’nchez, “Generalized Baum-Welch Algorithm Based on the Similarity between Sequences,” *PLoS One*, vol. 8, no. 12, p. e80565, 2013.

[132] R. C. Team and others, “R: A language and environment for statistical computing,” 2012.

[133] R. Killick and I. A. Eckley, “An R package for changepoint analysis,” *CRAN Repos.*, 2011.

[134] R. M. Stallman and G. N. U. E. Manual, “Free Software Foundation,” *El Proy. GNU--Fundaci{ó}n para el Softw. Libr.*, 1986.

[135] J. M. Chambers and T. J. Hastie, *Statistical models in S*. CRC Press, Inc., 1991.

[136] W. N. Venables and B. D. Ripley, *MASS: modern applied statistics with S*. 2002.

[137] R. N. Jorissen and M. K. Gilson, “Virtual Screening of Molecular Databases Using a Support Vector Machine,” *J. Chem. Inf. Model.*, vol. 45, no. 3, pp. 549–561, Apr. 2005.

[138] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen, “Active learning with support vector machines in the drug discovery process,” *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 2, pp. 667–673, 2003.

[139] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[140] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, “Feature extraction,” *Found. Appl.*, 2006.

[141] J. Fang, A. L. Varbanescu, H. Sips, L. Zhang, Y. Che, and C. Xu, “Benchmarking intel xeon phi to guide kernel design,” 2013.

[142] J. Jeffers and J. Reinders, *Intel Xeon Phi coprocessor high performance programming*. Newnes, 2013.

[143] J. Dean and S. Ghemawat, “MapReduce : Simplified Data Processing on Large Clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 1–13, 2008.

[144] Z.-H. You, J.-Z. Yu, L. Zhu, S. Li, and Z.-K. Wen, “A MapReduce based parallel SVM for large-scale predicting protein--protein interactions,” *Neurocomputing*, vol. 145, pp. 37–43, 2014.

[145] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Big data: The next frontier for innovation, competition, and productivity,” 2011.

ACRÓNIMOS

AlogP	Partition coefficient
AUC	Area Under the Curve
CI	Computational Intelligence
CMP	Chip Multiprocesor
CPSA	Charged Parcial Surface Area
CUDA	Compute Unified Device Architecture
CV	Crivado Virtual
DUD	Directory of Useful Decoys
ECP	Extended-Connectivity FingerPrints
EstCt	Estate counts
EstKy	Estate keys
FFP	Fragment FingerPrint
FLOPS	Floatingpoint Operations Per Second
GPB	Glucógeno Fosforilasa
GPGPU	General-Purpose Computation on Graphics Hardware
GPU	Graphics Processing Units
GPU	Graphics Processing Units
HPC	High Performance Computing
MC	Monte Carlo
MDA	Mean Decrease Accuracy
MDG	Mean Decrease Gini
MDLPK	MDL public keys
MolWe	Molecular Weight
MPI	Message Passing Interface
Mpola	Molecular polar surface area

MR	Mineralocorticoides receptor
Msolu	Molecular solubility
Msurf	Molecular surface area
NArRg	Number of aromatic rings
Natom	Number of atoms
NHAcc	Number of H-bond acceptors
NHDon	Number of H-bond donors
NNET	Neuronal Networks
NRing	Number of rings
NRotB	Number of rotatable bonds
OOB	Out Of Bag
PDB	Protein Data Bank
QSAR	Quantitative Structure–Activity Relationship
RNA	Red de Neuronas Artificiales
SIMD	Single Instruccition Multiple Data
SM	Streaming Multiprocesor
SP	Streaming Processor
SVM	Support Vector Machine
TK	Thymidine Kinasa
TMI	D-myo-inositol 3,4,5,6-tetrakisphosphate
VI	Variable Importance
VLSI	Very Large Scale Integration
VS	Virtual Screening