UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Integración en *Scipion* del *software* de *docking* molecular *Rosetta DARC* para su aplicación en descubrimiento y reposicionamiento de fármacos

Máster Universitario en Bioinformática y Biología Computacional

Autor: PARRA PÉREZ, Alberto Manuel

Tutor: SORZANO SÁNCHEZ, Carlos Oscar
Ponente: Roberto Marabini Ruiz

FECHA: junio, 2021

Agradecimientos

En primer lugar, me gustaría agradecer a Carlos Óscar Sorzano Sánchez y a José María Carazo García por haberme dado la oportunidad de formar parte de su grupo, por guiarme durante estos meses, por hacer posible este trabajo y, sobre todo, por la confianza que han depositado en mi.

Gracias a toda la Unidad de Biocomputación (BCU) del Centro Nacional de Biotecnología (CNB-CSIC) por acogerme y, en especial, a Roberto Marabini (mi ponente), a Blanca Benítez, Pablo Conesa, Marta Martínez y José Ramón por apoyarme y ayudarme siempre que lo he necesitado, por vuestra paciencia y por vuestra cercanía, aunque hayamos trabajado desde la lejanía.

Gracias a mi familia, por creer en mi, aguantarme y darme todas las facilidades que me han permitido estudiar lejos de casa. Sin ninguno de vosotros, y sin vuestro apoyo incondicional, nada de esto hubiera sido posible. Gracias a mis abuelos, Matilde y Manolo, porque, aunque ya no estéis a mi lado, aun me seguís dando fuerza y motivando para que sea, además de buen trabajador, buena persona. Gracias a mi yaya Carmen, por apoyarme cada día con sus llamadas y animarme a continuar mejorando. De igual forma, gracias a Jesús por haberme dado las fuerzas de seguir en aquellos momentos en los que estas flaqueaban.

Y, por ultimo, gracias a mis compañeras de piso, a Elena y a mis compañeros de Máster, por su apoyo y por hacer que estos dos años hayan sido mucho más divertidos.

Índice

Resumen

El cribado virtual basado en estructuras es una técnica in silico utilizada para filtrar grandes bases de datos de compuestos en busca de moléculas que tenga una actividad deseada sobre una diana proteica, reduciendo el conjunto de moléculas disponibles antes de realizar ensayos experimentales más costosos. Para ello, la técnica más usada es el acoplamiento entre proteína y ligando y la predicción de las energías de unión entre las dos. Esta técnica ha dado muy buenos resultados y ha permitido encontrar fármacos para tratar distintas enfermedades como el glaucoma, la hepatitis C y la hipertensión. Si bien las ventajas son evidentes, se producen con frecuencia un gran número de falsos positivos y falsos negativos debido a la complejidad de la predicción con precisión de la posición de unión correcta por la dificultad de parametrizar las interacciones de unión ligando-proteína. Para evaluar la unión existen distintas funciones de puntuación y distintos programas de acoplamiento como AUTODOCK, GLIDE y Rosetta DARC. Algunos funcionan mejor en casos específicos, pero no en casos más generales (bolsillos tradicionales), como aquellas dirigidas a realizar experimentos de docking en bolsillos superficiales de las proteínas implicados en interacciones proteína-proteína, como Rosetta DARC, programa elegido para su integración en la herramienta de flujos de trabajo, Scipion. Rosetta DARC, junto a distintas herramientas de preparación de proteínas y moléculas, se han integrado en forma de plugin para generar flujos completos de cribado virtual en esta herramienta. Se ha construido de forma que su uso sea sencillo e interoperable con otros programas que puedan ser integrados en Scipion para completar el flujo de trabajo en Scipion. Esta integración y programa se ha probado en dos casos de uso totalmente distintos: el macrodominio 1 (Mac1) de la proteína no estructural 3 (NSP3) del SARS-CoV-2, que presenta un bolsillo tradicional, y sobre la superficie de interacción entre NCS-1 y Ric8a, implicadas en el Síndrome X frágil.

Palabras clave: Acoplamiento molecular, Interacción proteína-proteína, Scipion, Integración de programas

Abstract

Structure-based virtual screening is an *in-silico* technique used to filter large databases of compounds for molecules that have a desired activity on a protein target, reducing the pool of available molecules before accomplishing more expensive experimental assays. For this, the most used technique is the protein-ligand docking and the prediction of its binding energies. This technique has given very good results and it has made possible to find drugs to treat different diseases such as glaucoma, hepatitis C and hypertension. While the advantages are obvious, large numbers of false positives and false negatives frequently occur due to the complexity of accurately predicting the correct binding position and the difficulty of parameterizing ligand-protein binding interactions. There are different scoring functions and different docking programs such as AUTODOCK, GLIDE, and Rosetta DARC to evaluate the dock. Some work better in specific cases, but not in more general cases (traditional pockets), such as those aimed at conducting docking experiments on surface pockets of proteins involved in protein-protein interactions, such as Rosetta DARC, the program chosen for its integration into the workflow tool, Scipion. Rosetta DARC, along with different protein and molecule preparation tools, have been integrated as a plugin to generate complete virtual screening flows in this tool. It has been built in such a way that its use is simple and interoperable with other programs that can be integrated into Scipion to complete the workflow in Scipion. This integration and program have been tested in two totally different use cases: macrodomain 1 (Mac1) of non-structural protein 3 (NSP3) of SARS-CoV-2, which presents a traditional pocket, and on the interaction surface between NCS-1 and Ric8a, implicated in Fragile X Syndrome.

Key words: Docking, Protein-protein interaction, Scipion, Software Integration

1. Introducción

Desde el comienzo de nuestra historia, los seres humanos hemos estado enfrentándonos a enfermedades y buscando remedios para ellas. Fue, durante el siglo XX y, especialmente en su segunda mitad, cuando se descubrieron la mayoría de los medicamentos que usamos actualmente, debido al avance en disciplinas como la biología, bioquímica y química orgánica, tanto en la mejor comprensión de la bases fisiopatológicas y moleculares de las enfermedades, como en el mayor conocimiento de las bases de la transformación química de moléculas, para mejorar sus propiedades y efectos [1]. Ahora, se ha hecho muy patente la importancia y necesidad del descubrimiento y reposicionamiento de medicamentos, entendiendo este último como el proceso de dar una nueva aplicación a un fármaco ya aprobado y comercializado, ya que estamos inmersos en la carrera de encontrar tratamientos y vacunas efectivas contra el virus SARS-CoV2.

Si bien es verdad que el número de fármacos aprobados por la FDA (del inglés, *U.S. Food and Drug Administration)* ha aumentado ligeramente en las últimas décadas (Figura 1A), debido al mejor conocimiento de la etiología de las enfermedades y el avance en la tecnología; aun el número de compuestos que alcanzan la clínica es bajo con respecto al número de compuestos probados en estudios iniciales, siendo el ratio de 1 de cada 10.000 compuestos [2]. De igual forma y contribuyendo a dicho ratio, las agencias reguladoras han incrementado los requisitos para la entrada de nuevos medicamentos al mercado (mayor eficacia y potencia, menor toxicidad, facilidad en la administración y asequibilidad), contribuyendo todo a que el coste medio, en la última década, para que un fármaco se comercialice sea aproximadamente de 965 millones de dólares, y que el tiempo promedio sea de 10 a 14 años, incluyéndose todas las fases de desarrollo de un fármaco (fase de descubrimiento y estudio de la diana terapéutica, fase de desarrollo y optimizado preclínico de distintas moléculas hasta candidatos a fármacos, fase de ensayos clínicos y fase de registro) (Figura 1B) [3].

En un primer momento, las empresas farmacéuticas, con el avance en el campo de la química combinatoria y el comienzo del uso del cribado de alto rendimiento (HTS, del inglés *High Throughput Screening*), mejoraron el proceso convencional de diseño y descubrimiento de fármacos, al acelerar y automatizar la selección de miles de compuestos simultáneamente. Sin embargo, esto tampoco salía rentable ya que es un enfoque de búsqueda costoso en cuanto a tiempo, coste económico, esfuerzo, falta de sensibilidad y fiabilidad, debido a la incertidumbre en el mecanismo de acción del compuesto. Además, las tasas de acierto suelen ser bajas y no muchos de los fármacos con efecto (compuestos *hits* y *leads*) pueden convertirse en candidatos reales y entrar en ensayos clínicos. Entonces, las compañías farmacéuticas tienden al uso de estrategias alternativas que evitan el uso de moléculas que, por su forma y propiedades predichas, no van a tener éxito. Por lo tanto, herramientas de descubrimiento de fármacos asistidas por ordenador, como el cribado virtual (denominándose también HTS virtual), está ganando popularidad como enfoque complementario al HTS convencional en la industria farmacéutica e investigación académica [4].

El **cribado virtual** es una técnica *in silico* que se utiliza para filtrar grandes bases de datos de compuestos en busca de moléculas activas potenciales, reduciendo el conjunto de moléculas antes de proceder a un HTS más costoso. Al no requerir la síntesis de compuestos, a diferencia del HTS, no está limitado por el espacio químico accesible experimentalmente. Otra diferencia que se encuentra en el cribado virtual es que se trata de una aproximación racional de diseño de fármacos ya que requiere información experimental, ya sea una estructura de proteína para el cribado virtual

basado en estructura (SBVS, del inglés *Structure-based virtual screening*) o un conjunto de compuestos activos conocidos, con las propiedades deseadas, para el cribado virtual basado en ligandos (LBVS, del inglés *Ligand-based virtual screening*) [5].

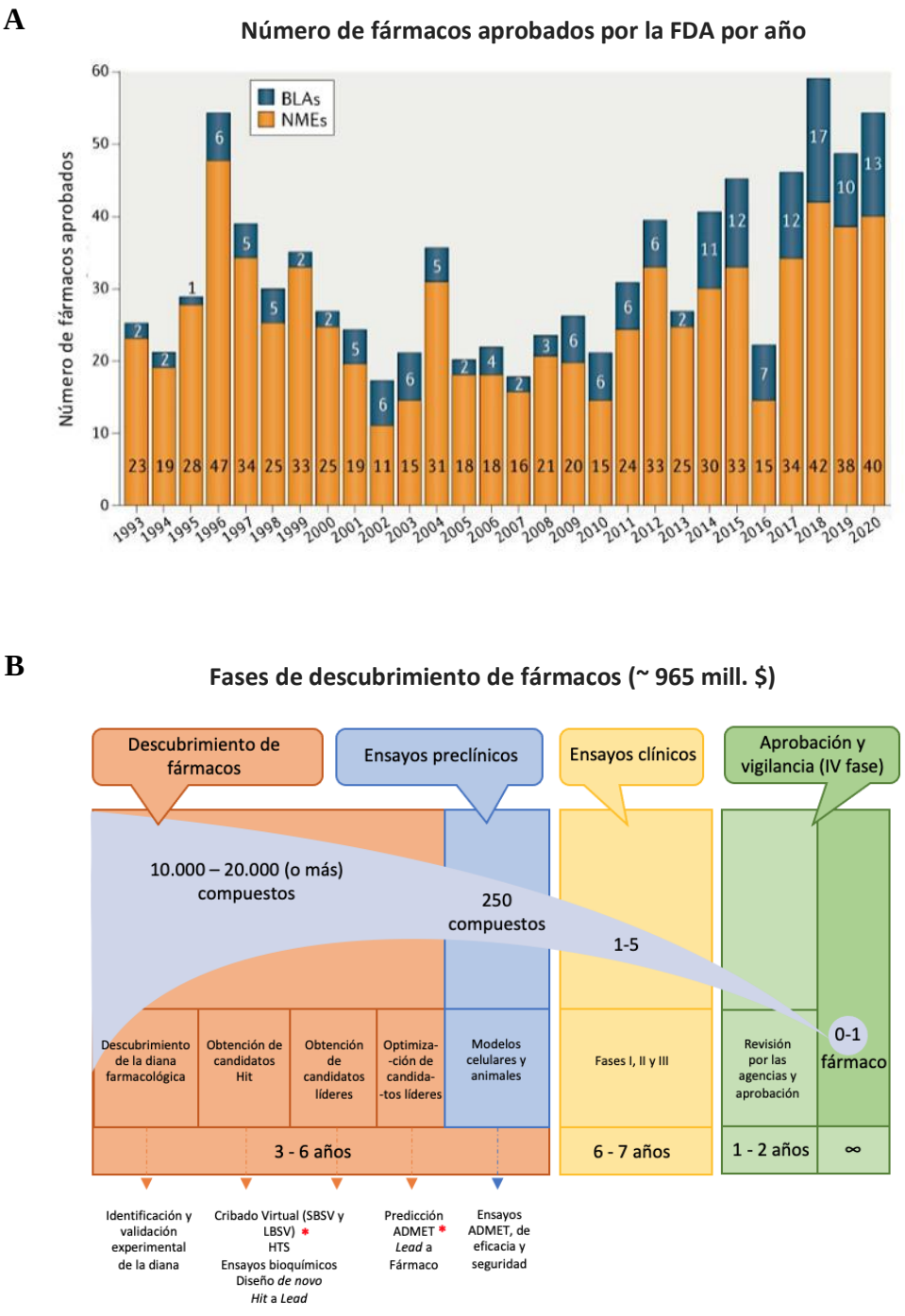


Figura 1. A) Número de fármacos aprobados por la FDA por año (Modificada de [2]). BLAs se refiere a las licencias de productos biológicos y NDAs a las solicitudes de nuevos medicamentos. **B)** Fases en el descubrimiento de fármacos, número de compuestos probados en cada fase, tiempo que se dedica normalmente a cada uno y técnicas experimentales y computacionales relevantes empleadas en la primera fase de descubrimiento de la diana farmacológica y obtención de un número de compuestos con las propiedades y función deseada sobre dicha diana.

Para ilustrar su uso, el cribado virtual ha contribuido a que algunos fármacos lleguen al mercado como el conocido captopril (medicamento antihipertensivo), la dorzolamida (usado para tratar el glaucoma), el zanamivir (un antivírico selectivo para el virus de la influenza), el boceprevir (inhibidor de la proteasa NS3 utilizado para el tratamiento de la hepatitis C y su causante, el virus de la Hepatitis C), entre otros [6].

En esta introducción explicaremos las dos aproximaciones de cribado virtual, LBVS y SBVS centrándonos más en esta última, ya que el programa (Rosetta DARC), que se quiere integrar en la plataforma *Scipion*, es un programa destinado a realizar *docking* molecular, técnica basada en la estructura de una proteína y el acoplamiento de ligandos a ella.

1.1. Cribado virtual basado en ligandos

El LBVS se basa en el estudio y conocimiento de un conjunto de compuestos que se unen activamente al sitio objetivo de una proteína de interés y resulta útil cuando no se dispone de la estructura tridimensional resuelta experimentalmente de dicha proteína o, la predicción *in silico* no es posible. Los métodos más populares dentro del LBSV son los estudios de similitud 2D de la estructura molecular, detección de moléculas candidatas en bibliotecas de compuestos utilizando farmacóforos (disposición o patrón de átomos, grupos funcionales en una molécula pequeña requerida para la interacción específica con su objetivo biológico y su actividad [7]), relación cuantitativa estructura-actividad (3D-QSAR) y predicción de las propiedades ADMET (Absorción, Distribución o transporte desde el lugar de administración hasta su lugar de acción, Metabolismo, Excreción y Toxicidad) de los compuestos [4].

A grandes rasgos, esta aproximación computacional consiste en seleccionar, y posteriormente analizar, aquellas moléculas con propiedades fisicoquímicas similares al farmacóforo de entre un gran repositorio. La ventaja principal es que las moléculas obtenidas tienen una estructura diversa pero una función similar. Este enfoque es útil para la optimización de moléculas líderes (*lead*), para obtener compuestos activos con actividad biológica a concentraciones nanomolar, de manera que LBVS y SBSV son en realidad enfoques complementarios cuando se dispone tanto de la estructura de la proteína diana, ya que, como veremos la estrategia SBSV es capaz de realizar un filtrado mucho mayor de compuestos mediante *docking* molecular, como de un conjunto de compuestos con la actividad deseada.

1.2. Cribado virtual basado en estructuras

Este método de cribado virtual se utiliza cuando está disponible la estructura tridimensional (propia y/o depositada en el *Protein Data Bank* (PDB)) de la macromolécula diana contra la que se quiere buscar un fármaco, que se una a ella de manera adecuada y que tenga las propiedades ADMET optimizadas.

El SBSV presenta una serie de ventajas e inconvenientes a tener en cuanta para su uso [8]. Entre las ventajas podemos encontrar:

I. Reducción del tiempo y del coste involucrados en el cribado de millones de compuestos.

II. No es necesaria la existencia física de la molécula, por lo que se puede probar computacionalmente incluso antes de sintetizarla.

III. Existe una gran variedad de herramientas y algoritmos disponibles para SBVS.

En cambio, las desventajas pueden ser las siguientes:

I. Algunas herramientas funcionan mejor en casos específicos, pero no en casos más generales [9], como aquellas dirigidas a realizar experimentos de *docking* en bolsillos superficiales de las proteínas implicados en interacciones proteína-proteína (PPI, del inglés *protein-protein interactions*)

II. Es difícil predecir con precisión la posición de unión correcta debido a la dificultad de parametrizar la complejidad de las interacciones de unión ligando-proteína.

III. Se producen con frecuencia falsos positivos y falsos negativos.

Los pasos para hacer un SBSV son los siguientes:

El primer paso (1) consiste en la identificación y validación en términos de resolución y calidad, de la o las proteínas diana. Las estructuras tridimensionales se determinan experimentalmente mediante técnicas de biología estructural como: cristalografía de rayos X, resonancia magnética nuclear (RMN), crio-microscopía electrónica (crio-ME), creciendo exponencialmente el uso de esta última en los últimos 5 años [10], y, si no fuera posible obtenerlas mediante técnicas experimentales, se recurre a los métodos de predicción estructural *in silico*, como el modelado por homología (se necesita como mínimo un 40% de similitud en la secuencia [11]), y el modelado *ab initio*.

En el segundo paso (2), el objetivo es identificar el bolsillo de unión, que suele ser una pequeña cavidad interior o una superficie mayor de la proteína, donde se quiere que los candidatos a fármacos se unan para producir el efecto deseado (ej. inhibir a una enzima o bloquear la interacción con otras proteínas, respectivamente) [11]. Antes de recurrir a predictores de bolsillos de unión, la primera opción en el descubrimiento racional de fármacos es llevar a cabo una revisión bibliográfica y buscar si ya existen estructuras con ligandos unidos a dichos bolsillos. Aun así, y a pesar de la naturaleza dinámica de las proteínas, existen métodos capaces de detectar los posibles residuos o regiones de unión para moléculas pequeñas y delimitar los bolsillos de unión, como *LigSite* [12], basado en el uso de la geometría de la proteína y la posibilidad de que moléculas de agua puedan ingresar en el interior de ella; *Q-SiteFinder* [13], basado en la energías de interacción de van der Waals (vdW) de un grupo metilo [11]; y *SiteMap*, de la plataforma Schrödinger [14], que combina las dos estrategias anteriores, incluyendo, además, medidas de hidrofobicidad.

El tercer paso (3), una vez identificado el bolsillo de unión, sería realizar experimentos de *docking* (o también un diseño de novo más complejo). De esta forma, se llevaría a cabo el cribado virtual, propiamente dicho, ya que se cogería una librería de compuestos con propiedades químicas similares a los fármacos y compuestos líderes, almacenados en bases de datos como ZINC15 (https://zinc15.docking.org) [15], con más de 100 millones de compuestos, en diversos formatos 2D (*smiles* o *smi*) y 3D (*mol2, pdb, sdf*), y se analizaría su acoplamiento contra la proteína diana. Este

cribado mediante *docking* se utiliza para encontrar y filtrar aquellas moléculas en función de su energía de unión o puntuación dada por las funciones de puntuación de cada método, para probar aquellos compuestos con mejores puntuaciones *in vitro* [11].

1.3. *Docking* molecular

El *docking* molecular es una técnica de simulación virtual de interacciones moleculares. Esta es la técnica más popular en SBSV ya que predice la conformación y unión de ligandos dentro de un sitio objetivo en una macromolécula. Se puede aplicar a estudios de fenómenos moleculares, tales como la unión de un ligando e interacciones intermoleculares en un complejo y su estabilidad.

Los algoritmos de *docking* predicen las energías de unión y clasifican los ligandos por medio de varias funciones de puntuación. La selección de los ligandos con una unión apropiada depende de dos factores: (1) el estudio o barrido de un gran espacio conformacional de los ligandos, que define las posibles formas de unión, mediante algoritmos de búsqueda, y (2) la predicción de la energía de unión de cada conformación, usando funciones de puntuación. Este proceso se realiza de forma recursiva hasta converger a una solución de mínima energía, en el que la unión del ligando se evalúa mediante esas funciones de puntuación [7].

En cuanto al primer factor (1), hay dos formas de realizar docking molecular: ***docking* con ligando flexible** y ***docking* con proteína flexible**, de acuerdo con si se hacen variaciones de la estructura de los ligandos o de la proteína.

1.3.1. Algoritmos de búsqueda

En el *docking* con ligando flexible, existen tres tipos de algoritmos distintos para estudiar la flexibilidad del ligando y hacer un barrido del espacio conformacional. Estos algoritmos de búsqueda conformacional pueden ser clasificados en **métodos sistemáticos**, **métodos estocásticos** y **métodos deterministas basado en simulación de dinámica molecular** (DM), correspondiendo este último también al *docking* con proteína flexible:

I. **Métodos de búsqueda sistemática (algoritmos codiciosos)**. Generan ligeras variaciones en los parámetros estructurales, cambiando gradualmente la conformación de las moléculas y generando confórmeros. El algoritmo va a calcular la energía asociada al espacio conformacional y, después de numerosos ciclos de búsqueda y evaluación, suele converger a la solución de mínima energía correspondiente al modo de unión más probable. Aunque el método es eficaz para explorar el espacio conformacional, puede converger a un mínimo local. Este problema puede superarse realizando búsquedas simultáneas a partir de diferentes puntos del espacio energético mediante bibliotecas pre-generadas de conformaciones distintas [16].

II. **Métodos estocásticos**. La búsqueda conformacional se lleva a cabo modificando aleatoriamente los parámetros estructurales de los compuestos. Estas modificaciones se aceptan o rechazan dependiendo de métodos y funciones de probabilidad y energía, como los

métodos basados en algoritmos genéticos y en el método de Monte Carlo, evitando caer en mínimos locales. Sin embargo, el costo computacional asociado a ellos suele ser mayor que para los algoritmos sistemáticos [11].

III. **Métodos deterministas basado en DM**. Por último, en cuanto al *docking* frente a un ligando o/y proteína flexible, lo que se suele hacer son simulaciones de DM, que es una técnica para estudiar el comportamiento dinámico de las macromoléculas ya que los sistemas biológicos no son estáticos. Esta estrategia, muy costosa computacionalmente, rara vez se usa para propósitos de cribado virtual y, solo se usa para la optimización de compuestos líderes. Algunos algoritmos disponibles para minimizar la energía en DM son el método de Newton-Raphson, el método *steppest descent*, los métodos de mínimos cuadrados y de gradiente conjugado [11].

Además, cada programa de *docking* molecular (Tabla 1) incorpora alguno de estos métodos de búsqueda conformacional en sus algoritmos. Aquellos como FRED [17], DOCK [18] o GLIDE [19] usan métodos sistemáticos, usando algoritmos de construcción incremental del ligando para evitar el problema de la explosión combinatoria, al aumentar las posibles conformaciones del ligando cuanto mayor sean sus grados de libertad (enlaces simples a rotar); y, por ejemplo, AUTODOCK [20], PLANTS [21] o GOLD [22], utilizan métodos estocásticos basados en algoritmos genéticos [10, 15].

1.3.2. Funciones de puntuación

La segunda parte necesaria de los programas de *docking* molecular es la función de puntuación, mencionada anteriormente. Esta función es muy relevante ya que el cribado virtual dará como resultado un gran número de complejos proteína-ligando acoplados, y la evaluación y clasificación de estas interacciones moleculares predichas se convierte en el paso crucial para la identificación de aquellos candidatos más probables a fármaco. Las funciones de puntuación se utilizan para estimar la fuerza de las interacciones no covalentes entre un ligando y el bolsillo o lugar seleccionado de la macromolécula mediante distintos métodos matemáticos heurísticos [8].

Al ser estas funciones las responsables de predecir la afinidad de unión, mediante el cambio de energía libre de *Gibbs* de unión (ΔG) , entre una diana y un ligando, son las principales responsables del éxito o fracaso de los programas de docking, ya que aunque su uso está muy extendido, dicha estimación no es trivial (desde 1990 se estima que se han publicado más de 100 funciones de puntuación [4]).

Estas funciones se dividen en cuatro clases generales [9], más funciones que combinan las 4 principales:

I. **Funciones basadas en la física o en los campos de fuerza**. Estiman la energía libre de unión sumando las fuerzas de vdW intermoleculares, interacciones electrostáticas y enlaces o puentes de hidrógeno entre todos los átomos de los dos socios del complejo diana-ligando. También se tienen en cuenta las contribuciones de la solvatación y la entropía en funciones más completas, ya que en las ecuaciones de mecánica cuántica de los campos de fuerza (ej. AMBER, GROMOS, OPLS). Algunos ejemplos de programas de docking que utilizan este tipo de funciones son AUTODOCK [20], DOCK [18], GOLD [22] (GoldScore) [16].

II. **Funciones empíricas o basadas en regresión**. Se basan en la agrupación de varios tipos de interacciones entre proteína-ligando, como las fuerzas hidrófobas, fuerzas de vdW, enlaces de hidrógeno y entropía o el número de enlaces simples inmovilizados en la formación del complejo. Se emplea un método estadístico como la regresión lineal múltiple para ajustar los coeficientes de la función de puntuación y obtener un valor para cada interacción. Estas funciones han demostrado realizar una buena predicción para muchos complejos proteína-ligando. AUTODOCK (incluye varias funciones de puntuación distintas y la posibilidad de utilizar propias) [20], GLIDE (GlideScore) [19] y DOCK6 [23] son algunos ejemplos de programas y funciones de puntuación [16].

III. **Funciones basadas en el conocimiento**. Para estimar una puntuación utilizan observaciones estadísticas de contactos intermoleculares en complejos receptor-ligando con conformaciones estructurales conocidas, basándose en el hecho de que grupos funcionales o ciertos tipos de átomos que interactúan con frecuencia son energéticamente favorables y contribuyen a la afinidad de la unión. Por ejemplo, algunas funciones de esta categoría son DrugScore [24] y SMoG [25] [16].

IV. **Funciones basadas en descriptores o aprendizaje automático (ML)**. Estas funciones utilizan propiedades fisicoquímicas y biológicas de los compuestos, las proteínas y de los patrones de interacción. Estas propiedades se codifican como variables y se utilizan métodos de ML (ej. *Random Rorest*, *support-vector machines* y redes neuronales) para derivar modelos matemáticos que estimen una puntuación que describa la unión [4]. Últimamente, el desarrollo de este tipo de funciones se está volviendo popular ya que, 1) estos métodos toman en consideración implícitamente las interacciones entre un ligando y un objetivo mientras que 2) ignoran las interacciones propensas a errores. Además, 3) pueden dar lugar a funciones con una dependencia no lineal entre las interacciones de enlace, dando mejores resultados que las que usan modelos lineales. Algunas funciones que usan esta estrategia son NNScore [26] y RF-Score [27] [8].

V. **Funciones híbridas que combinan distintas aproximaciones**. Estas funciones combinan dos o más de las categorías de funciones de puntuación mencionadas anteriormente. Esto se debe a que cada aproximación tiene sus ventajas e inconvenientes y el uso combinado puede suplir las limitaciones de cada uno y realizar una mejor estimación de la energía de interacción de la molécula pequeña con la macromolécula [16]. Algunos ejemplos son la función que utiliza GalaxyDock [28] o DARC de la suite Rosetta [29], [30] (función de puntuación de forma y *Ref2015* en la minimización[31]), que ambas combinan la aproximación basada en la física (la primera por usar fuerzas electroestáticas) y en campos de fuerza, y basadas en el conocimiento.

Combinando, entonces, algoritmos de búsqueda conformacional y funciones de puntuación aparecen multitud de programas destinados a hacer *docking* (Tabla 1) no solo en bolsillos interiores de las proteínas sino también en zonas superficiales implicadas, normalmente, en interacciones proteína-proteína, aunque también pueden ser centros activos superficiales, como el caso de la PETasa, enzima capaz de degradar polímeros de plástico (PET), *Ideonella sakaiensis* [32].

Entre los programas de *docking*, aquellos que más se suelen utilizar para hacer cribado virtual son los que aparecen en la Tabla 1, recogiendo qué tipo de licencia, algoritmo de búsqueda y función de puntuación tienen.

Tabla 1. Programas más comunes usados en cribado virtual

Programa	Licencia	Algoritmo de búsqueda	Función de puntuación	Referencia
AUTODOCK4	Gratis para uso académico	Estocástico (Algoritmo genético)	Híbrida (campos de fuerza y empírica)	[20]
DARC	Gratis para uso académico	Estocástico (*Particle Swarm Optimization*)	Conocimiento y *ref2015* (híbrida entre campos de fuerza y conocimiento)	[29], [33]
DOCK6	Gratis para uso académico	Sistemático (ajuste mediante esferas)	Empírica	[23]
FLEXX	Comercial	Sistemático (generación incremental)	Empírica	[34]
FRED	Comercial	Sistemático (búsqueda exhaustiva)	Híbrida (conocimiento y empírica)	[17]
GALAXYDOCK	Gratis	Estocástico (Algoritmo genético)	Híbrida (campos de fuerza y conocimiento)	[28]
GLIDE	Comercial	Sistemático (búsqueda exhaustiva)	Empírica	[19]
GOLD	Comercial	Estocástico (Algoritmo genético)	Campos de fuerza	[22]
ICM	Comercial	Estocástico (Monte Carlo)	Campos de fuerza	[35]
PLANTS	Gratis para uso académico	Estocástico (A*nt Colony Optimization*)	Empírica	[21]
rDOCK	Gratis	Estocástico (Algoritmo genético y minimización con Monte Carlo)	Híbrida (campos de fuerza y conocimiento)	[36]
SWISSDOCK/ EADOCK DSS	Gratis para uso académico	Sistemático (búsqueda localizada)	Campos de fuerza	[37]

Tabla 1. Programas de *Docking* Proteína-Ligando más usados para tareas de cribado virtual. En la tabla se muestra qué tipo de licencia tiene cada programa (gratuito, gratuito solo para uso académico o comercial (de pago) en todos los casos), qué tipo de algoritmo de búsqueda conformacional y función de puntuación usan cada uno de ellos.

1.4. *Docking* proteína-ligando en sitios de interacción de proteínas

La eficacia del descubrimiento de fármacos dirigidos al proteoma ha disminuido en los últimos años, al contrario que aquellos dirigidos al interactoma, es decir, dirigidos a evitar las interacciones proteína-proteína (PPI) [38]. Las PPI juegan un papel fundamental en la mayoría procesos biológicos y multitud de enfermedades, desde las rutas metabólicas y vías de señalización celulares, como por ejemplo aquellas implicadas en la regulación de la proliferación celular (MDM2/p53 y c-Myc/Max), de la apoptosis o "muerte celular programada" (Bcl-2/Bax) y del estrés oxidativo celular (Keap1/Nrf2) [39], que este último conlleva, de forma general, a la muerte celular en enfermedades inflamatorias y Alzheimer y la creación de un microambiente propenso al desarrollo de un tumor; hasta la comunicación y reconocimiento entre células y proteínas de sistema inmune, como los anticuerpos, la infección por patógenos y la propia y "simple" formación de estructuras cuaternarias y complejos poli-proteicos, entre otros muchos ejemplos de la participación y relevancia de las PPI.

La investigación actual sobre el interactoma humano sugiere que la cantidad de PPI ronda entre 240.000 descritas y 650.000 interacciones estimadas, aunque posiblemente haya más (entre las más de 20.000 proteínas descubiertas) [40] y solo una pequeña cantidad de ellas han sido utilizadas como objetivos farmacológicos.

Los programas de *docking* han tenido éxito en muchos de los casos en el que la diana de los fármacos era un bolsillo tradicional proteína-ligando (ej. centro activo de una enzima) de una proteína. Sin embargo, su rendimiento se ve afectado en las superficies de PPI porque estas presentan cinco características fisicoquímicas principales y diferentes a los bolsillos tradicionales (Figura 2): desde un punto de vista geométrico, (1) la superficie de interacción es relativamente grande, (2) plana y (3) la forma tiende a fluctuar al unirse. Además, (4) las interacciones entre las dos superficies que van a unirse están formadas por átomos hidrófobos, diferentes de los medicamentos tradicionales dirigidos al bolsillo tradicionales, hidrofílicos. Por último, (5) las PPI normalmente no tienen moléculas naturales que se unan a ellas, por lo que un diseño basado en ligandos es un reto, al igual que el estructural [38].

Para el desarrollo de fármacos que actúen sobre PPI, se han aplicado con éxito métodos de docking convencionales como GOLD, GLIDE e ICM [41]–[43], a menudo en combinación con la compatibilidad con farmacóforos. Estos métodos se usaron porque las interfaces usadas tenían bolsillos donde podían caber ligandos. Sin embargo, para aumentar el rendimiento de la búsqueda de fármacos (descubrimiento o reposicionamiento) se están desarrollando métodos que aborden esta problemática, como, por ejemplo, DARC [29], perteneciente a la suite de Rosetta, ZDOCK, FRODOCK, HAWKDOCK y GALAXYPEPDOCK [38].

La motivación principal de este Trabajo de Fin de Máster (TFM) es incorporar al panel de métodos de Scipion, *framework* que permite la ejecución de flujos de trabajo, un programa de *docking* que sea específico e idóneo, en tiempo, eficacia y coste computacional, para llevar a cabo un procedimiento de cribado virtual completo en sitios de interacción de proteínas. De entre los métodos destinados a esto, se eligió Rosetta DARC, debido al método que usa para mapear la superficie de interacción, usando para ello vectores sobre la superficie y su capacidad de usar GPU y paralelizar el cribado de grandes repositorios de moléculas.

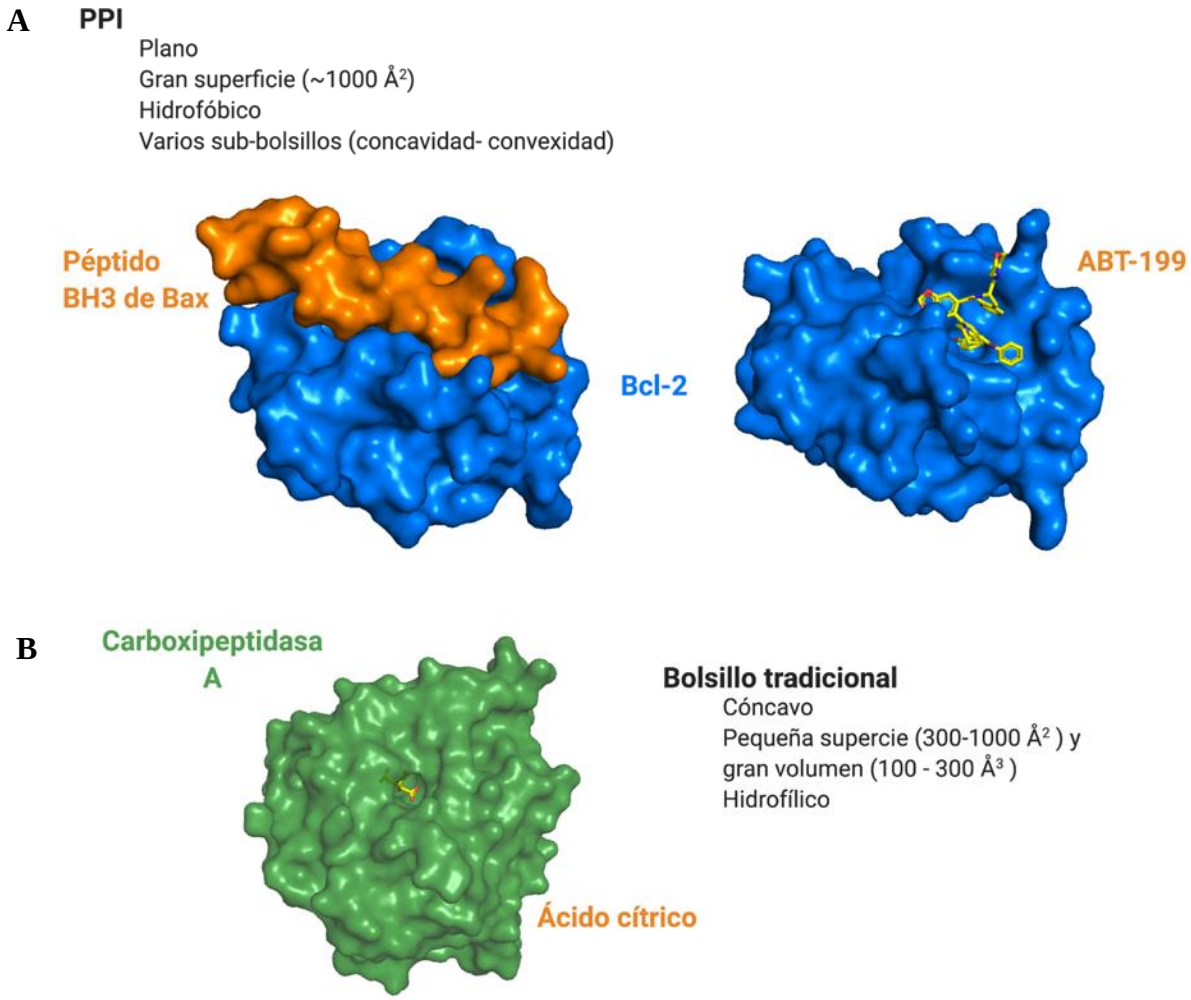


Figura 2. Diferencia entre la forma y las propiedades de los bolsillos superficiales involucradas en las PPI (A) y los bolsillos tradicionales (B). **A)** Estructura de la proteína anti-apoptótica Bcl-2 en complejo con el péptido BH3 de Bax, proteína pro-apoptótica, que uniéndose a Bcl-2 impide la apoptosis (*Protein Data Bank* (PDB): 2XA0). La estructura de la derecha corresponde a Bcl-2 unido al ligando ABT-199 (PDB: 6GL8), que bloquea la interacción con Bax y promueve la apoptosis en células tumorales. **B)** Carboxipeptidasa A unidad al ácido cítrico, sustrato de dicha enzima (PDB: 3KGQ)

2. Objetivos

El éxito experimental de los programas de cribado virtual es limitado debido a la alta tasa de falsos positivos. Esto se debe principalmente a tres factores: 1) los programas de cribado virtual optimizan una única función de score para predecir la unión entre un ligando y la proteína, 2) los flujos de análisis de datos en cribado virtual son actualmente relativamente simples, poco transparentes en su uso y no interoperables entre los distintos programas y 3) la mayoría de los programas se destinan a realizar un cribado sobre bolsillos tradiciones en las proteínas.

De esta forma el **objetivo general** de este Trabajo de Final de Máster (TFM) es la creación de un *plugin* para el *framework Scipion* que integre el programa de *docking* pensado para bolsillos superficiales de proteínas, Rosetta DARC, que permita la realización de un proceso de cribado virtual de moléculas candidatas a fármacos o ligandos de una proteína dada, de una forma simple, interoperable y trazable.

Los objetivos parciales para la consecución del objetivo final son los siguientes:

- **Objetivo 1**. Estudio de la estructura y el funcionamiento de *Scipion*.
- **Objetivo 2**. Integración en *Scipion* de Rosetta DARC, programa de *docking* para realizar un proceso de cribado virtual pensado para regiones de PPI, aunque no exclusivamente. Creación de los protocolos y objetos necesarios para la preparación de las estructuras de proteínas, para la preparación del conjunto de pequeñas moléculas que se quieren testar en el cribado virtual y para el docking molecular.
- **Objetivo 3**. Integración de paquetes y programas, como OpenBabel, y diseño de un flujo de trabajo en *Scipion*, que aseguren la interoperabilidad y la conversión de formatos para Rosetta DARC y otros programas que se incorporarán en un futuro como AUTODOCK.
- **Objetivo 4**. Aplicación de los flujos de trabajo diseñados a dos casos de uso de cribado virtual.

3. Integración de Rosetta DARC en el *framework Scipion*

Para la integración del programa de *docking* molecular Rosetta DARC es necesario conocer cuál es su funcionamiento y cuáles son sus parámetros de entrada y salida, incluidos sus formatos. Así mismo, es necesario conocer cómo es la herramienta donde se va integran Rosetta DARC, *Scipion*, su estructura y la funcionalidad que da a los paquetes que se desarrollan e integran en ella.

3.1. Rosetta DARC

El paquete de software Rosetta [30] (https://www.rosettacommons.org) está disponible gratuitamente para uso académico e incluye algoritmos para el análisis y modelado de estructuras de proteína y para el *docking* de ligandos tanto en bolsillos tradicionales (Rosetta LIGAND), como en zonas superficiales, es decir, *docking* entre dos proteínas (Rosetta DOCK) o entre proteína y ligandos (Rosetta DARC (del inglés, *Docking Approach using Ray-Casting*)).

En concreto, el programa que se integrará en el *framework Scipion*, Rosetta DARC, es un programa diseñado principalmente para hacer *docking* en bolsillos planos y poco profundos, como los descritos en la introducción, basado en el uso de "*ray-casting*". Rosetta DARC es un método de *docking* flexible para los ligandos y rígido para la proteína diana, ya que va a realizar una búsqueda de la conformación más favorable, en términos de energía de unión, de los ligandos que se acoplen en la estructura. Este método, principalmente se basa en la comparación entre la forma "observada" de un bolsillo desde un punto situado dentro o fuera de la proteína (fijado previamente) con la forma "observada" de un compuesto desde el mismo punto de vista, solapando ambas topologías en el sitio de unión (Figura 3A) y puntuando dicha complementariedad de forma usando la Fórmula (1) [29].

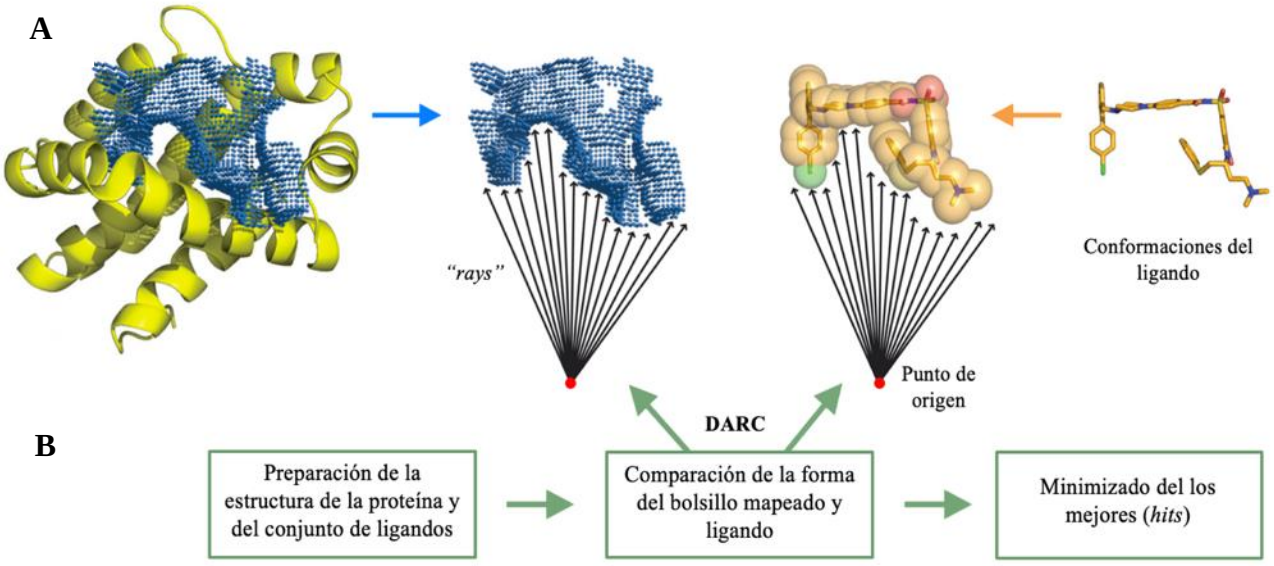


Figura 3. Método "*ray-casting*" usado por Rosetta DARC. **A)** Primero se generan rayos o vectores desde un origen dentro de la proteína (punto rojo) y se mapea la topografía del bolsillo al calcular la intersección de estos rayos con el bolsillo (izquierda). Para puntuar la complementaria de forma de un ligando, DARC proyecta los mismos rayos y desde el mismo origen (derecha). Si el ligando (en dicha conformación y posición) es perfectamente complementario de forma al bolsillo, cada vector intersectará con el ligando a la misma distancia que con el bolsillo mapeado. **B)** Esquema del flujo de trabajo de cribado virtual con Rosetta DARC. Figura adaptada del artículo que muestra la segunda versión de DARC, DARC 2.0 [33].

Además, este método, al definir la forma del bolsillo mediante el uso de vectores, como veremos a continuación, no permite una variación conformacional de la proteína cuando se une el ligando, proporcionando un punto de partida adecuado para los casos en los que la estructura de la proteína diana no sea perfectamente óptima para el ligando. Esto puede darse en aquellos casos en los que se espera que pequeños cambios conformacionales de la proteína se produzcan como consecuencia de la unión del ligando. Se puede decir, por tanto, que se trata de un método de baja resolución, que permite detectar, de forma más amplia, posibles ligandos al no focalizar en detalles sino en la forma [29] y, posteriormente, en la complementariedad de cargas [33]. Comparándolo con otros programas (DOCK [18], PLANTS [21], AUTODOCK [20]), y al ser un método de baja resolución, Rosetta DARC superaba la tasa de verdaderos positivos de estos programas cuando la conformación de la proteína no era la optima. Es decir, no presentaba en la estructura usada para el *docking* un ligando unido con un quimiotipo similar a los compuestos probados [29], por lo que el bolsillo no estaba bien definido por cambios conformacionales en la proteína.

El flujo de trabajo típico de un proceso de cribado virtual con Rosetta DARC (Figura 3B) se puede dividir en 3 etapas principales: preparación de la estructura de la proteína y la librería de compuestos que se usaran frente a la estructura, el docking con DARC (que engloba varias etapas de preparación de sus parámetros de entrada) y un filtrado posterior del listado de compuestos, ordenado por la función de puntuación del programa, usando para ello la minimización *fullatom* de Rosetta [31]. Los pasos concretos y específicos del cribado con DARC se explicarán con detalle (con formatos de ficheros y variables modificables en el uso de los programas), durante la descripción de la integración de Rosetta DARC en Scipion. A grandes rasgos, el flujo sería (Figura 4):

1) Preparación de la estructura de tridimensional de la proteína:

    I. Eliminar moléculas de agua.

    II. Eliminar los HETATM (heteroátomos) en la estructura de la proteína (formato pdb).

    III. Eliminar cadenas redundantes de la proteína. Rosetta DARC trabaja con una sola cadena.

    IV. Añadir los átomos de hidrógeno mediante modelado (de forma general, las estructuras procedentes de cristalografía de rayos X y crio-microscopía electrónica (crio-ME) no permiten resolver la posición de los hidrógenos debido a la resolución de las estructuras) y átomos que puedan faltar en cadenas laterales. Estos átomos serán necesarios posteriormente para asignar las cargas parciales y generar una caja o *grid* de potencial electroestático, que se usará para estudiar la complementariedad entre las cargas de bolsillo y el ligando.

2) Preparación del conjunto de moléculas, en formato mol2 y pdb, que se quieren utilizar durante el *docking*:

    I. Establecer las cargas parciales de los distintos átomos que forman la molécula

    II. Generar confórmeros de cada ligando. Los confórmeros son estereoisómeros que pueden cambiar su orientación espacial, convirtiéndose en otro isómero de la misma molécula, a temperatura ambiente, por rotación en torno a enlaces químicos simples.

3) Generar la *grid* electroestática (formato agd) para la proteína completa (disponible en DARC2.0 ya que permite tener en cuenta la complementariedad de cargas entre la proteína y el ligando).

4) Generar el fichero de rayos o vectores (formato pdb y txt) que modela la forma del bolsillo, preferiblemente, superficial, aunque también puede ser interno, localizando 1o 2 residuos (residuos diana) incluidos en dicho bolsillo o flanqueantes de la región. Para generar el bolsillo, primero se selecciona un origen a 30 Å (o varios en DARC 2.0) desde el residuo diana seleccionado en la dirección del centro de masa de la proteína. Desde este origen, los rayos se proyectan a través de puntos en la superficie del bolsillo (Figura 3A) [33]. Según cómo estos rayos intersecten con el bolsillo y el ligando, se calculará el valor de la función de puntuación (1) que da una comparación de forma del *docking* proteína-ligando.

5) *Docking* con Rosetta DARC. Para ello, es necesario cargar en Rosetta DARC una estructura o conformación de la proteína, con los ficheros de *grid* electroestática y topología del bolsillo, y el conjunto de ligandos, cada uno con su lista de confórmeros generados previamente. Para llevar a cabo el proceso de docking es necesario tener en cuenta que tiene función de puntuación de "superposición" basada en el conocimiento (Fórmula (1)) y, otro sumado en la función que tiene en cuenta las en fuerzas electroestáticas (segundo sumatorio de la Fórmula (1)) [33].

De esta forma, se define la "puntuación de la superposición" de DARC como el grado en que coinciden las topografías o formas del bolsillo y del ligando, sumando todos los rayos de la siguiente manera. Además, se incluye el sumatorio que tiene en cuenta la complementaria de cargas incluido en DARC2.0:

$$\sum_{rayos} \begin{Bmatrix} c1 * (\rho_{ligando} - \rho_{bolsillo}) & si & \rho_{ligando} > \rho_{bolsillo} \\ c1 * (\rho_{bolsillo} - \rho_{ligando}) & si & \rho_{ligando} < \rho_{bolsillo} \\ c3 & si & los\ rayos\ no\ cruzan\ con\ el\ ligando \\ c4 & si & los\ rayos\ no\ cruzan\ con\ el\ bolsillo \end{Bmatrix} + c5 * \sum_{\substack{átomos\ del \\ ligando}} q_i \phi_i \quad (1)$$

dónde $\rho_{bolsillo}$ y $\rho_{ligando}$ son la distancia a la que un rayo determinado se cruza con el bolsillo y el ligando, respectivamente. Para determinar el potencial electrostático (ϕi) en la ubicación de un átomo de ligando dado (i), se usa una interpolación trilineal de los puntos de la *grid* generada sobre la proteína más cercanos que engloban cada átomo del ligando. Las *qi* se refieren a las cargas atómicas parciales. Los parámetros c1, c2, c3, c4 y c5, son valores positivos, que se han optimizado utilizando un conjunto de datos de entrenamiento de inhibidores de la PPI conocidos y las propias regiones de PPI a los que se unen [29][33].

Desde un punto de vista físico, la *condición 1* se aplica a los vectores que alcanzan el bolsillo antes que el ligando, indicando un empaquetamiento insuficiente entre la proteína y el ligando; en cambio, la *condición 2* se aplica a los vectores que alcanzan el ligando antes que al bolsillo; lo que indica un choque estérico. La *condición 3* penaliza los vectores que no se cruzan con el ligando, es decir, en este caso el que el ligando es pequeño en relación con el bolsillo. El último término es para aquellos rayos que no intersectan con el bolsillo, lo que indica que el ligando es demasiado grande. Por lo

tanto, para cada conformación unida en un bolsillo proteico se evalúa la complementariedad de forma entre el ligando y la superficie del bolsillo [29], de manera que a mayor puntuación, menor es la complementariedad de forma y carga.

Por otro lado, cualquier algoritmo de *docking*, como hemos visto anteriormente, necesita de un algoritmo de búsqueda conformacional. Los confórmeros, que se evaluaran de forma independiente mediante la función de puntuación (Fórmula (1)), se generan debido a que Rosetta DARC los utiliza en su **algoritmo estocástico de búsqueda conformacional** y optimización basado en un algoritmo evolutivo, en concreto, en el **algoritmo PSO** (del inglés, Particle Swarm Optimization) [29], que suele converger más rápido al mínimo global y conformación optima que los algoritmos genéticos.

El método PSO es un método de optimización aleatorio basado en el comportamiento poblacional de aves y/o las interacciones sociales humanas, de manera que cada individuo, solución o partícula, tienden a tener las mismas características o reunirse ("enjambre o *swarm"*) en ciertos puntos de un espacio de búsqueda dado (energético o de algún tipo de característica o puntuación), afectándose entre sí las características de cada partícula individual al recorrer dicho espacio. Este método se ha aplicado con éxito en varias aplicaciones distintas al *docking* ya que su campo de aplicación principal es la optimización continua de una función real [44] y por sus características se adapta bien al problema del *docking*, donde cada conformación del ligando tiene que minimizarse con respecto a una función de puntuación.

De esta forma, Rosetta DARC utiliza PSO para evaluar la posición y la orientación de los confórmeros aportados ("partículas") y optimizar la función de puntuación (Fórmula (1)). Este enfoque permite que la posición y orientación de cada confórmero o partícula se adapte en respuesta a las otras partículas, moviéndose hacia las partículas locales y globales con mejor puntuación con un cambio fijado previamente (tamaño de la conversión o el paso) que depende de las puntuaciones relativas de las partículas. Después de varias iteraciones, en las que todas las partículas se desplazan en el entorno energético en respuesta unas a otras, el "enjambre" de partículas converge idealmente en la solución óptima global (en este caso, el confórmero con mejor ajuste y de menor puntuación); o, si se ha alcanzado el número máximo de interacciones, la búsqueda de la conformación mínima se termina [45]. El confórmero seleccionado de un ligando para un conjunto es aquel que tiene menor puntuación DARC.

El posible inconveniente del método Rosetta DARC es la necesidad de "comunicación" entre los distintos confórmeros, pudiendo impedir la obtención de confórmeros en varias máquinas independientes. Sin embargo, en el caso de DARC (y enfoques de cribado virtual que utilizan algoritmos genéticos), la función de puntuación se puede evaluar con la suficiente rapidez como para que la simulación de todas las soluciones candidatas (partículas) pueda evaluarse razonablemente en un solo procesador. Además, en un contexto de cribado virtual, ejecutar cada miembro del conjunto de compuestos a cribar como un trabajo independiente aún puede permitir la paralelización en varias máquinas [45].

6) Minimización energética de los mejores resultados (normalmente el 10% mejor). Los resultados obtenidos en Rosetta DARC sirven como punto de partida y cribado más rápido para una mayor optimización utilizando la función de puntuación *ref2015* [31], basada en el campo de fuerza de todos los átomos (minimización *full-atom*). Finalmente, estos complejos minimizados se reordenan sobre la base de consideraciones energéticas (por ejemplo, energía

de interacción) así como consideraciones estructurales (por ejemplo, número de grupos polares en regiones hidrófobas). Los compuestos de puntuación mejores pueden ser candidatos para una caracterización adicional en ensayos bioquímicos y ensayos *in vitro* en células.

Los autores ampliaron DARC [29] a su versión posterior, DARC 2.0. [33], que es la que se integra en Scipion. Las características de esta nueva versión y que mejoran la anterior son:

1. El lanzamiento de rayos se realiza desde cuatro puntos de origen adicionales a 45° del eje del punto de origen, capturando más información topográfica. Estos puntos adicionales no se seleccionarían por el usuario.

2. Se añadió el término de la Expresión (2) que incorpora complementariedad electroestática, incluido ya en la Fórmula (1). El potencial electrostático se estableció en cero en el interior de la proteína y en un valor muy desfavorable fuera del sitio de unión definido para penalizar a los candidatos que abandonan el sitio de unión:

$$\sum_{\substack{\text{átomos del} \\ \text{ligando}}} q_i \phi_i \qquad (2)$$

3. Se permite el uso de GPU, acelerando la generación de la forma del bolsillo y la realización del docking y el minimizado del mismo.

En resumen, la idea interesante de los algoritmos de Rosetta DARC es el uso de *ray casting*, que es adecuado para capturar características geométricas de superficie plana en interfaces PPI.

3.2. SCIPION

Scipion (http://scipion.i2pc.es) es una herramienta o *framework* de ejecución de flujos de trabajo y protocolos, modular (basada en *plugins* o complementos que aportan una funcionalidad nueva a la herramienta), distribuido para sistemas operativos Linux y desarrollada por la Unidad de Biocomputación del Centro Nacional de Biotecnología (CNB-CSIC). Fue concebido para llevar a cabo el procesamiento de imágenes procedentes del campo de la microscopía electrónica tridimensional (3DEM) y solucionar los problemas de integración e interoperabilidad entre los diferentes paquetes orientados a la reconstrucción de volúmenes o mapas de densidad de moléculas a partir de las imágenes obtenidas mediante 3DEM [46].

En los últimos 5 años, los avances en crio-ME, y en concreto en crio-ME de una sola partícula, han hecho posible obtener estructuras de macromoléculas biológicas con una resolución casi atómica [47]. Como ha aumentado en gran medida los mapas de crio-ME de resoluciones altas (*resolution revolution*) se ha fomentado el uso y desarrollo de herramientas de modelado de estructura de proteínas capaces de producir modelos atómicos de alta calidad a partir de dichos mapas. Al igual que ocurre con los programas de procesamiento de imágenes, surgen muchos problemas prácticos al combinar diferentes paquetes en un mismo flujo de trabajo, dificultando el uso de estas herramientas de los no expertos y reduciendo su utilidad.

De esta forma, *Scipion* se ha extendido mediante la incorporación de programas para el trazado de modelos atómicos de proteínas derivados de mapas de crio-ME y la validación de dichos modelos. De esta forma, *Scipion* permite construir flujos completos de trabajo para la obtención de estructuras de proteínas usando crio-ME, desde el procesamiento de imágenes hasta la obtención de los mapas y la estructura atómica tridimensional y la validación de dichos mapas y estructuras [48].

En la mayoría de los casos, los paquetes que llevan a cabo el procesamiento de imágenes contienen cientos de pequeños programas que realizan tareas específicas y tienen entradas y salidas bien definidas, con sus propios formatos. Además, el principal objetivo de *Scipion* es integrar y gestionar la interacción entre varios paquetes y programas orientados a la reconstrucción de modelos tridimensionales de moléculas utilizando las imágenes de la ME, como por ejemplo XMIPP, RELION, EMAN, SPIDER, estando estos cuatro paquetes destinados al análisis de las imágenes y la obtención de los mapas y su refinamiento; EMRIGER de PHENIX y MOLPROBITY para la validación del ajuste entre mapa de crio-EM y modelo estructural y la validación geométrica de este último, CHIMERA para visualización de mapas de crio-ME y estructuras, entre muchos otros [29, 31]. Todos los paquetes tienen ventajas y desventajas, por lo que generalmente es necesario combinar los algoritmos de diferentes paquetes para obtener los resultados deseados y Scipion lo consigue.

Uno de los problemas principales que surgen al utilizar diferentes paquetes para una aplicación determinada, como el *docking* o el propio análisis de imágenes crio-ME, es la falta de estandarización en el formato de los archivos de entrada y salida necesarios para llamar a un programa determinado. Por ejemplo, si se quiere utilizar AUTODOCK, las proteínas y ligandos deben estar en formato pdbqt y la *grid* se encuentra en formato gpf, donde se señala cada archivo que conforma dicha *grid*, y los archivos en formato map; en cambio para usar Rosetta DARC, los ligandos deben estar en formato mol2 o pdb, según el paso en el que se usen (preparación o *docking*) y la *grid* en formato agd. Esto suele complicar la tarea del usuario, ya que debe ser consciente de las diferencias entre los flujos de trabajo y los resultados que proporciona cada paquete para poder combinarlos correctamente para obtener un resultado adecuado.

Con esta premisa y por la similitud en cuanto a la cantidad de programas disponibles, *Scipion* se convierte en la herramienta idónea para la integración de paquetes de softwares usados para el mismo propósito y para el manejo de los diferentes formatos de salida de los archivos generados por los diversos paquetes. Por ello, en este trabajo se utilizará para la integración de programas de preparación de proteínas y pequeñas moléculas y de *docking* molecular, siguiendo con la lógica de obtención de estructura de proteínas y su aplicación para SBVS y *docking*.

En *Scipion*, la gestión de los distintos formatos de entrada y salida se realiza a través de objetos especiales y conversiones en el interior de los *plugins* y protocolos, como aquellos que definen la estructura de proteínas (*AtomStruct*), conjuntos de pequeñas moléculas en distintos formatos (*SetOfSmallMolecules*), las *grid* electroestáticas (*gridAGD*), etc. Estos objetos se utilizan para hacer coincidir y suplir los requisitos de entradas y salidas de cada programa, por lo que la interacción entre ellos ya no es un problema del usuario. Además de la función integradora de *Scipion*, este *software* también permite realizar una trazabilidad de los flujos de trabajo ejecutados ya que guarda las entradas y salidas de cada paso ejecutado en el flujo de trabajo en una carpeta diferente y crea un archivo de registro (archivo *log*). Esto permite al usuario examinar los resultados y los posibles errores o advertencias que pueden aparecer durante la ejecución de dicho para aislar los pasos que conducen a un problema determinado.

Scipion trabaja mediante proyectos que puede crear el usuario para tener una independencia en cuanto los trabajos y análisis realizados de otros proyectos. También es posible importar y exportar un proyecto ya existente guardado en el ordenador para modificarlo o incluir nuevos pasos en él.

Scipion consta de la siguiente estructura y regiones importantes en la interfaz gráfica o GUI (del inglés, *Graphical User Interface*) (Figura 4):

1. La ventana situada a la izquierda de la GUI (Figura 5, cuadro azul) contiene todos los programas integrados en *Scipion* para la reconstrucción de modelos tridimensionales y la realización de *docking* con Rosetta DARC y AUTODOCK, así como para el manejo de objetos. Se pueden organizar de diferentes formas para facilitar la búsqueda de un algoritmo específico. Cuando se selecciona un programa, aparecerá una ventana emergente que muestra lo que se necesita (ficheros, objetos y parámetros) para ejecutar el programa. La Figura 4 muestra un ejemplo de esta ventana.

2. La ventana central (Figura 4, cuadro naranja) muestra el flujo de trabajo de un proyecto de cribado virtual, usando la representación de grafo. Los diferentes pasos del flujo (nodos) de trabajo se especifican mediante bloques que se insertan haciendo clic en cualquier programa presente en la ventana izquierda y completando las entradas. Las líneas que conectan los bloques (ramas) se utilizan para establecer el bloque que envía sus salidas o resultados como entradas de otro bloque. De hecho, Scipion y, en concreto cada proyecto, se almacena en forma de una base de datos *SQLite* que almacena la información de los protocolos, sus entradas y salidas y las rutas relativas a las carpetas y ficheros.

   Los bloques se pueden mostrar en cuatro colores diferentes. Un color verde significa que el bloque se ha ejecutado sin errores; un color rojo significa que la ejecución se detuvo debido a un error que apareció durante la ejecución; un color naranja indica que el programa se está ejecutando; y un color azul, indica que el programa está pendiente de ser ejecutado conforme finalicen los protocolos precedentes.

   Al hacer doble clic en cada bloque, la ventana del protocolo que se muestra en la Figura 7 aparecerá nuevamente. Esto permite al usuario ejecutar nuevamente un programa dado o cambiar los parámetros de entrada para obtener un mejor resultado. Así mismo, se permite copiar cada uno de los bloques y ejecutarlos como copia, para tener distintas ejecuciones del mismo programa, con distintos parámetros, de forma sencilla.

3. La ventana situada en la parte inferior (Figura 4, cuadro verde) se utiliza principalmente para mostrar las salidas del bloque que se está analizando. Tiene tres pestañas diferentes: La primera pestaña muestra un resumen del proceso realizado y los archivos de entrada y salida del bloque que se pueden visualizar haciendo clic en el botón "*Analyze Results*"; la pestaña "*Méthods*" se utiliza para mostrar el archivo de registro generado por *Scipion* para cada bloque; y la última pestaña muestra el archivo "*stdout*" que contiene toda la información sobre la ejecución del

bloque, incluyendo los mensajes que indican el error que detuvo la ejecución. Esta pestaña se actualiza en tiempo real durante la ejecución del bloque.

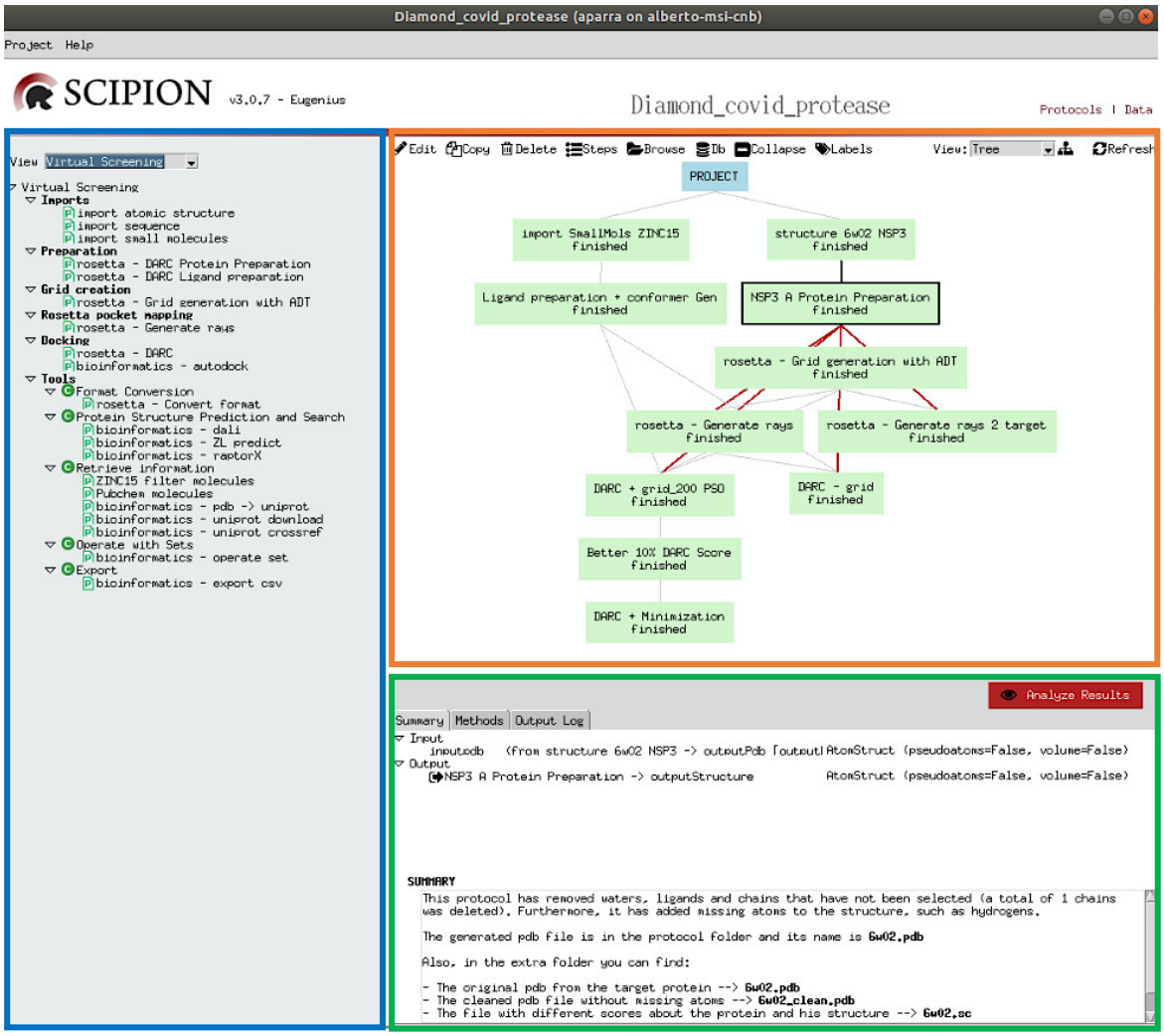


Figura 4. GUI de *Scipion*. A la izquierda (azul) se muestran los protocolos disponibles para configurar un flujo de trabajo. En este caso se muestran aquellos disponibles para realizar un cribado virtual, aunque dispone de muchos otros que se podrían ver en las distintas secciones del visor de la parte superior. En el parte central (naranja) se encuentra el flujo de trabajo realizado, siendo cada bloque el protocolo, coloreado en verde porque ha terminado su ejecución con éxito, y las líneas que lo conectan las entradas y salidas de cada protocolo. En la parte inferior (verde) se muestra la sección que resumen la ejecución del protocolo, los métodos usados y el archivo de *log* de la propia ejecución.

También es importante mencionar que *Scipion* se compone principalmente de dos tipos diferentes de programas:

- Protocolos, Visores y *Wizards*: Estos programas están escritos en Python y se dedican principalmente a gestionar la ejecución de los algoritmos incluidos en los diferentes paquetes. También son responsables del manejo del formato de las entradas y las salidas y se engloban en cada uno de los bloques que se pueden usar y ver en la ventana central de Scipion.

- Librerías: Estos programas están escritos en C ++ y se encargan de la gestión de los proyectos y la GUI. Contienen los algoritmos que luego serán llamados por los protocolos para realizar una tarea específica.

3.3. Integración de Rosetta DARC

Rosetta DARC se ha integrado en *Scipion* en la forma de complemento o *plugin*, estando escrito completamente en Python, siguiendo el esquema de la Figura 5. Este se encuentra alojado en el repositorio de GitHub, *Scipion-chem-rosetta* (https://github.com/scipion-chem/scipion-chem-rosetta).

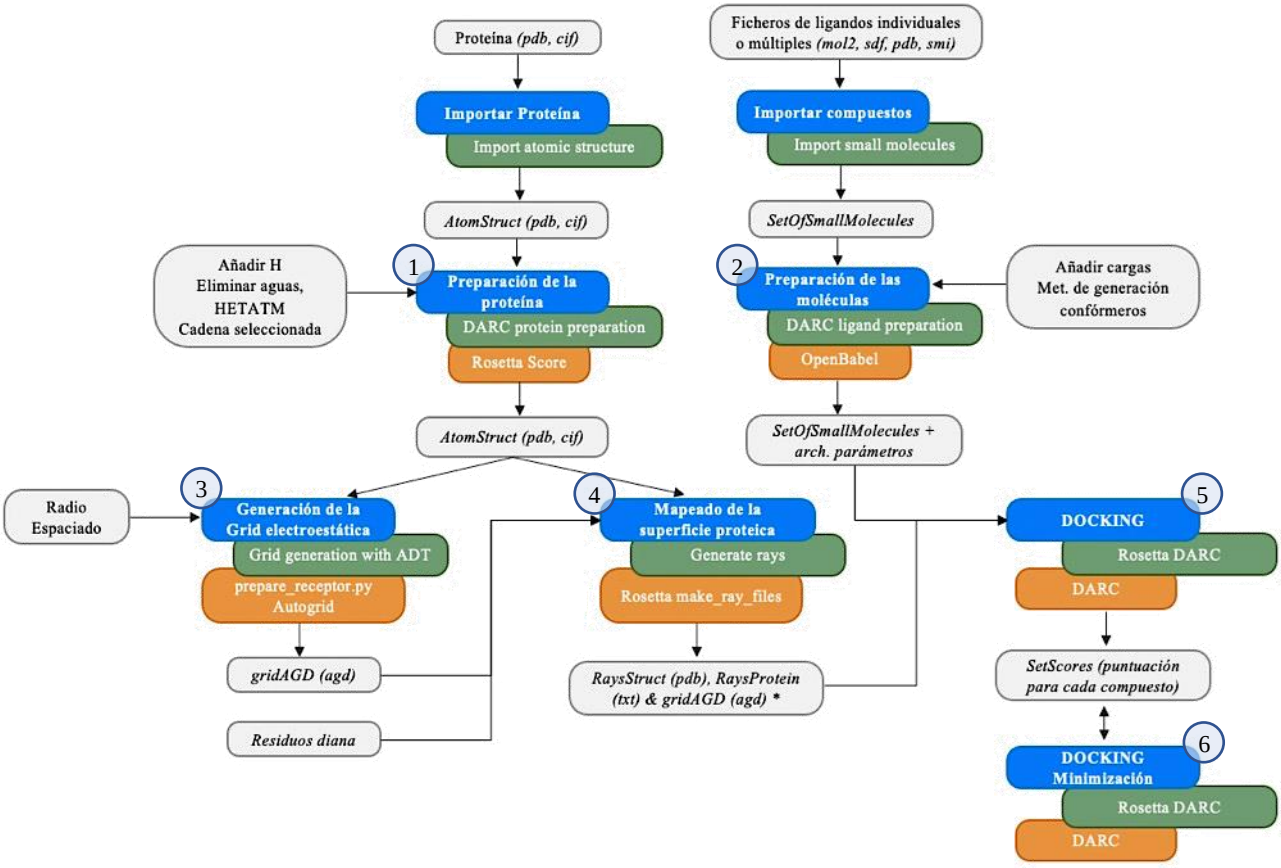


Figura 5. Esquema del flujo de trabajo integrado en *Scipion* para el uso de Rosetta DARC en cribado virtual. Un esquema similar puede verse en la Figura 5 donde se muestra la GUI de Scipion. En este esquema se muestran las distintas etapas (cajas azules) en el cribado virtual con DARC (1-Preparación de la proteína, 2-Preparación de los ligandos, 3-Generación de la grid electroestática, 4-Generación del archivo de rays que mapea la superficie o bolsillo del ligando, 5-*Docking* con DARC y 6-Minimización de los mejores resultados en el *docking*). En verde se muestra el nombre de los protocolos integrados en *Scipion* para llevar a cabo este flujo al completo, así como las entradas y salidas generales de estos protocolos como objetos de *Scipion*. Así mismo, en naranja están reflejados los paquetes y programas usados e integrados.

Actualmente, el repositorio contiene aquellos protocolos, visores y *wizards* que están relacionados específicamente con el programa de *docking* en PPI, Rosetta DARC, aunque ha sido diseñado para poder incorporar más programas procedentes del paquete Rosetta, ya que, el puntero del *plugin* está dirigido a la ruta del paquete Rosetta y no hacia los programas específicos.

El *plugin* está organizado en módulos y submódulos de Python, los cuales tienen los archivos y scripts necesarios para su correcta instalación en Scipion, así como, todos aquellos protocolos que permitirán llevar a cabo el proceso de cribado virtual comentado en la introducción. La estructura del *plugin*, además, sigue la convención establecida para los módulos de Python (Figura 6):

A continuación, se describen los archivos y submódulos que deben contener los *plugins* de Scipion:

3.3.1 Módulo principal del *plugin* de Rosetta

Ficheros relacionados con la descripción, requerimientos e instalación del *plugin* a través de PyPI, como un módulo de Python. Entre estos ficheros, los más importantes son:

README.rst:. Contiene la información acerca del contenido del *plugin*, así como los pasos a seguir para su correcta instalación y uso. Es necesario seguirlos cuidadosamente ya que para descargarse e instalarse el paquete Rosetta, en primer lugar, se necesita una licencia. Cuando se tenga el paquete descargado y compilado, se podría proceder a la instalación del *plugin* en Scipion, para que así se reconozca la localización del paquete Rosetta.

requirements.txt: Se indican los módulos que son necesarios para el correcto funcionamiento de los protocolos del *plugin*, que se instalarán de forma automática al instalarlo en *Scipion*. En el caso de que no se instalaran, habría que hacerlo desde sus repositorios de GitHub. Se necesitarán *scipion-em* (*plugin* básico en Scipion que define objetos y protocolos para ME, que reutilizamos para las funciones de *docking*), *scipion-pyworkflow* (proporciona toda la base para crear protocolos, visores y *wizards*) y *scipion-em-bioinformatics* (*plugin* de reciente creación que contiene objetos útiles como el que define la información acerca de las moléculas usadas en el proceso de cribado virtual, *SetOfSmallMolecules*).

LICENSE: Licencia de uso del *plugin*. Se usa la licencia *GPL-3.0*, que permite el uso libre de los programas y su adaptación a las necesidades de cada usuario. También permite distribuir copias, mejorar el programa y hacer públicas esas mejoras.

setup.py: Se trata del script de compilación del *plugin*. La función *setup* creará el paquete para cargarlo en PyPI. *setup* incluye la información sobre el *plugin* como el nombre, su número de versión, qué otros paquetes son necesarios para los usuarios (recogidos en el *requirements.txt*) y qué submódulo o carpeta (rosetta) contiene la información de instalación, protocolos, visores, *wizards*, etc.

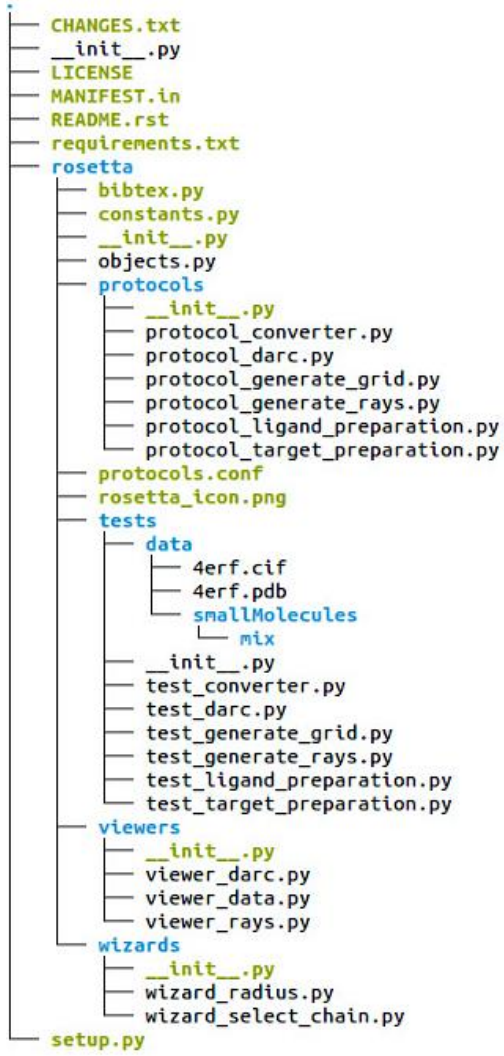


Figura 6. Árbol de directorios y ficheros del plugin creado, *Scipion-chem-rosetta*.

3.3.2. Submódulo Rosetta del *plugin* de Rosetta

Contiene una serie de submódulos con los protocolos, test de dichos protocolos y visores y wizards. Además, contiene otros archivos que se describirán a continuación. Dentro de cada carpeta, y al tratarse de submódulos de Python, se necesita de un archivo *__init__.py*, que se utiliza para enumerar las clases que contienen cada uno de los módulos y submódulos en Python. De esta forma, las clases definidas en cada archivo de Python estarán disponibles a nivel de *plugin*.

Los archivos más relevantes contienen el módulo de Rosetta son:

3.3.2.1. __init__.py

Este *script* se va a ejecutar en primer lugar cuando se instala el módulo en el intérprete de Python de *Scipion*, y en él se define el entorno, variables y rutas hacia los programas externos (Rosetta y OpenBabel) que usarán los protocolos del *plugin*. Esta definición se hace a través de la clase Plugin, que contiene una serie de métodos de clase importantes:

defineVaribles: Define variables que usará *Scipion* y que se encuentran escritas en el fichero su configuración, *scipion.conf*. Las variables que se definen son: ROSETTA_HOME, que guarda la ruta absoluta del paquete Rosetta en nuestro ordenador, y OPENBABEL_ENV_ACTIVATION, que guarda el comando de activación del ambiente de *conda* para el uso de las funciones de OpenBabel.

OpenBabel [49] es un paquete que contiene programas útiles en quimioinformática, que permiten convertir, filtrar y manipular archivos con formatos utilizados en típicamente en esta área (pdb, cif, smi, sdf, mol2, etc.). Además, contiene programas para construir confórmeros, comparar estructuras químicas y buscar similitudes entre ellas, entre otras.

getRosettaDir: Realiza una búsqueda en árbol de directorios y archivos del ordenador, buscando el directorio del paquete Rosetta, para cualquiera de sus versiones (se recomienda tener la 3.12 o superior). Si encuentra algún paquete instalado, devuelve la ruta completa al paquete para incorporarla a la variable ROSETTA_HOME en *scipion.conf*. En caso contrario, no devuelve ninguna ruta y el usuario debería añadirla al fichero de configuración.

getProgram: Función que devuelve la ruta absoluta a uno de los *scripts* de Rosetta, que se usarán en los protocolos. Para ello, la entrada de esta función es la ruta al paquete de Rosetta, la ruta al subdirectorio que contiene el programa que se desea usar y el nombre de dicho programa. La segunda y tercera entrada se definen en el archivo de constantes (*constants.py*) del *plugin*.

runRosettaProgram: Función que permite lanzar a la terminal de Linux una línea de comando con la ruta del programa de Rosetta que se desea usar con una serie de argumentos, que se definen en los protocolos correspondientes. Además, se puede establecer un directorio de trabajo para ejecutar dicha línea de comando.

addopenbabelPackage: Función que lleva a cabo la instalación de OpenBabel a través de *conda*.

runOPENBABEL: Al igual que el método *runRosettaProgram*, permite lanzar una línea de comando en Bash ejecutando OpenBabel en su forma *obabel*.

validateInstallation: Función que comprueba si la instalación del protocolo se ha producido de manera correcta, mediante la comprobación de la existencia de la ruta que se ha guardado en la variable ROSETTA_HOME. En el caso que ésta no exista, se le indica al usuario que debe añadirla al archivo de configuración ya mencionado.

3.3.2.2. constants.py

Contiene todas las constantes que se importan al resto de ficheros Python como los protocolos y el *__init__.py*. Las constantes definidas se corresponden a las rutas a distintos directorios relevantes en Rosetta y los nombres de cada uno de los programas usados para la integración del método DARC.

3.3.2.3. objects.py

Objetos usados para almacenar entradas o salidas de los protocolos del *plugin*. En concreto, los objetos creados derivan de clases ya creadas en la base de *Scipion* como es el caso de los objetos *EMFile*, que se refiere a la ruta de un fichero; *AtomStruct*, que almacena la estructura tridimensional de una macromolécula u objeto que pueda representarse en formato pdb o cif/mmcif; *EMObject*, que permite almacenar, para una entrada dada, distinta información a modo de diccionario; y *EMSet*, que almacena en una base de datos *SQLite* distintos *EMObject*, con los mismos campos.

3.3.2.4. Protocols

Los protocolos van a permitir llevar a cabo todo el flujo de trabajo de cribado virtual con Rosetta DARC. Para cada uno de ellos se mostrará el formulario creado, con las entradas requeridas obligatoriamente, junto a las opcionales y avanzadas. Además, se explicarán cada una de ellas para darle formato de guía de uso a este TFM.

Los protocolos se crean siguiendo la lógica de programación basada en objetos, creando métodos o funciones de una clase, que deriva normalmente de la clase *EMProtocol* de *Scipion* y contiene funciones comunes usadas en los protocolos, para la ejecución de un programa deseado. Existen algunas funciones obligatorias que contienen todos los protocolos:

- _defineParams: Esta función permite crear un formulario en la GUI de Scipion con todos los parámetros de entrada que se necesitan para que el protocolo funcione de manera adecuada.

- _insertAllSteps: Esta función se inicia una vez que se le da al botón de "*Execute*" de los formulares y contiene una llamada por cada función que se quiere ejecutar en el protocolo, incluyendo condiciones de ejecución según los parámetros de entrada. Estas funciones se crean como métodos de clase y son las que proporcionan la funcionalidad deseada al protocolo.

3.3.2.4.1. DARC Protein Preparation

En este protocolo se aborda la preparación de la estructura tridimensional de la proteína. Los parámetros de entrada se muestran en el formulario del protocolo (Figura 7) y son los siguientes:

- Estructura tridimensional de una proteína, en forma de objeto *AtomStruct*, obtenido en un protocolo previo llamado *import atomic structure*. De esta forma, haciendo una búsqueda en los objetos de Scipion con la lupa (🔍), podemos seleccionar la estructura que se desea preparar. Este objeto contiene la ruta a un fichero de estructura en formato cif o pdb, que internamente será transformado al formato de trabajo estándar para proteínas de Rosetta DARC, el pdb.

- Booleano para indicar si se añaden a la estructura proteica hidrógenos u otros átomos de cadenas laterales no trazados en la estructura.

- Booleano para eliminar las posibles moléculas de agua que estén incluidas en el archivo de estructura.

- Booleano para eliminar los ligandos unidos a la estructura y otros heteroátomos (HETATM) no pertenecientes a la estructura atómica de la proteína

- Booleano para eliminar las cadenas redundantes de la proteína. En caso afirmativo aparecerá un parámetro de entrada para seleccionar la cadena de la proteína sobre la que se quiere realizar el cribado virtual. Esta cadena se selecciona mediante un *wizard* (*wizard_select_chain.py*) (Figura 7, derecha) que va a recorrer el archivo pdb y va a extraer información sobre las distintas cadenas que contiene la proteína.

- Booleano que indica la posibilidad de utilizar una semilla constante para el uso del programa de Rosetta, *score.static.linuxccrelease* (nombre que recibe en sistemas operativos Linux), que añade los hidrógenos y otros átomos de cadenas laterales, además de calcular cual sería la puntuación de la estructura de la proteína basada en la función *ref2015*, que tiene en cuenta las interacciones, fuerzas electroestáticas, enlaces intramoleculares y geometría de la macromolécula. De esta forma, sería constante el resultado del programa, tomando importancia en el protocolo que realiza el *docking*, DARC. En caso afirmativo, se establece por defecto la semilla 1111111, pudiendo cambiarse.

Con estos parámetros, el protocolo, en primer lugar, va a recorrer el fichero pdb (el formato cif se convierte a formato pdb con una función importada desde el módulo pwem de *Scipion*, *cifToPdb*) y extrae solo aquellas líneas ATOM o HETATM, más la cabecera del pdb, que se hayan indicado a través de los booleanos. En segundo lugar, se lanza una línea de comando a la terminal de Linux para el uso del programa *score.static.linuxccrelease* que genera una serie de ficheros, almacenados en la carpeta destinada al protocolo, y que se usarán para construir la salida del protocolo.

El resultado del protocolo es un archivo en formato pdb con la estructura de la proteína procesada como se indicado a través del formulario y un objeto *AtomStruct*, que apunta a dicho archivo.

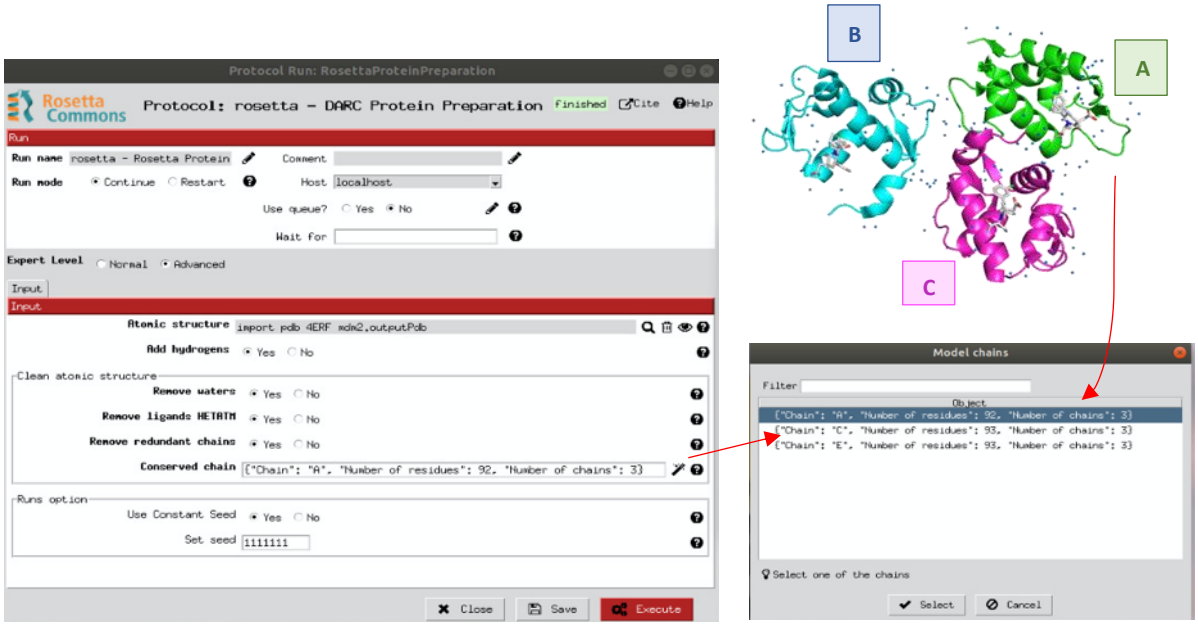


Figura 7. Formulario del protocolo *DARC Protein Preparation* con los parámetros de entrada (izquierda) y el *wizard* para seleccionar la cadena de la proteína a preparar añadida en el formulario (PDB: 4ERF, proteína MDM2 que se une con p53 y bloquea su acción de control del ciclo celular), en el que se selecciona la cadena A (derecha).

3.3.2.4.2. Grid generation with ADT

En DARC2.0, la función de puntuación puede tener en cuenta la complementariedad de carga entre la conformación del ligando y el sitio de unión seleccionado en la proteína. Para ello es necesario crear una matriz tridimensional o *grid*, que represente el valor de carga para cada punto de una caja que engloba a la proteína.

El primer problema para la generación de dicha *grid* es que Rosetta DARC depende del formato agd (formato OpenEye ASCII), que consiste en una cabecera de 5 líneas, que contiene información acerca de las dimensiones de la grid, y, a continuación, un número de líneas igual al valor del diámetro de la *grid* (Figura 8, derecha*)*, dividido por la distancia en Angstroms (Å) que separa a cada punto en la *grid*, elevado al cubo, para que la matriz cuadrada y simétrica. Este formato, como su propio nombre indica, solo puede generarse utilizando uno de los programas *Listings* (preferiblemente *Listings_2*, que calcula los valores de carga asociados a cada coordenada de la *grid*) de la librería de programas ZAP ToolKit de OpenEye (www.eyesopen.com). El problema surge cuando, para poder usar las herramientas creadas por OpenEye, se necesita una licencia de pago.

Para solventar este programa y poder utilizar en Rosetta DARC, además de la complementariedad de forma, la complementariedad de cargas electroestáticas, se aprovechó la forma de generar la *grid* electroestática de AutoDockTools (ADT) [20], software de uso libre, que permite, además de generar la *grid* de cargas electroestáticas, asignar las cargas atómicas parciales a la macromolécula utilizando el método de cargas de *Marsilli-Gasteiger*, basado en la electronegatividad de los átomos [50], necesarias para nuestro propósito.

Para generar la *grid* usaremos la función de ADT, *prepare_receptor4.py* y AUTOGRID, que al igual que para Rosetta, desde el plugin *Bioinformatics* de *Scipion* (https://github.com/scipion-em/scipion-em-bioinformatics), se instala y tiene funciones para llamar y ejecutar programas de AUTODOCK. Por lo tanto, este protocolo depende de dicho *plugin*.

Los parámetros de entrada son (Figura 8):

- Estructura tridimensional de la proteína, ya preparada, en forma de objeto *AtomStruct*, obtenida con el protocolo anterior, *DARC Protein Preparation*.

- El valor del radio (Å) que engloba a la estructura de la proteína desde su centro de masas. Para seleccionar este valor se utiliza un *wizard* creado por David Herreros, estudiante predoctoral de la Unidad de Biocomputación del CNB, usando la librería *matplotlib* . Este wizard se utiliza para generar máscaras en estructuras (https://github.com/scipion-em/scipion-em/blob/devel/pwem/wizards/wizards_3d/mask_structure_wizard.py) ya que proporciona, entre otros datos, el valor del radio de dicha máscara (Figura 8, derecha).

- El valor de la separación entre cada punto de la matriz tridimensional o *grid* electroestática, que por defecto es 0,375 Å.

Con estos parámetros de entrada, el flujo de trabajo del protocolo es el siguiente:

1. Asignar las cargas a los átomos presentes en la estructura de la proteína y generar un archivo en formato pdbqt, con la función *prepare_receptor4.py*, que será el que use AUTOGRID para generar la *grid* que se necesita.

2. Generar el archivo de parámetros de AUTOGRID (GPF), utilizando los siguientes parámetros, que se obtienen a partir de los parámetros de entrada del protocolo:

    - Número de puntos en cada una de las tres dimensiones de la *grid*. Estos valores idénticos se obtienen dividiendo el valor del radio entre la distancia de separación entre cada punto de la matriz.

    - Espaciado entre cada punto de la matriz tridimensional

    - Centro de masa de la proteína, obtenido utilizado para ello el método *centerOfMass* de la clase *AtomicStructHandler* de la librería de *Scipion pwem*.

3. Ejecutar AUTOGRID. Éste genera una serie de ficheros entre los que se encuentra uno, con la terminación "*e.map*" que hace referencia a la *grid* electroestática de la proteína. Este fichero es idéntico al formato en formato *agd* que necesita Rosetta DARC, cambiando exclusivamente la cabecera.

4. Crea un objeto *gridAGD* que apunta hacia el fichero en formato *agd* creado en el paso anterior. Este será la salida del protocolo, que tendrá que usarse en el siguiente protocolo de generación de los vectores que mapean la superficie de *docking* y en que lo realiza.

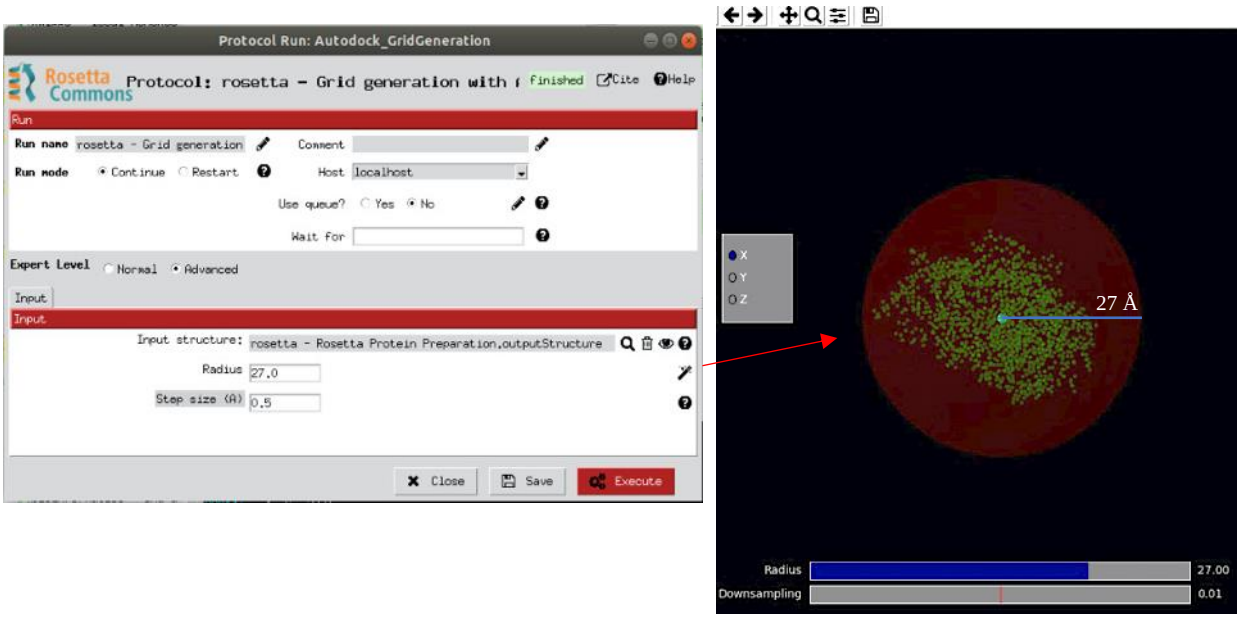


***Figura* 8** Formulario del protocolo *Grid generation with ADT* con los parámetros de entrada (izquierda) y el *wizard* para seleccionar el radio de la cadena de la proteína seleccionada previamente (PDB: 4ERF, cadena A), mostrando los átomos como puntos en verde y la esfera roja indicaría la longitud de la *grid*, en cuanto al diámetro, usado para establecer las dimensiones de la *grid* (derecha).

3.3.2.4.3. Generate rays

El objetivo de este protocolo es generar el fichero de vectores que modela la forma del bolsillo (superficial o tradicional) donde se quiere hacer *docking* y que servirá para comparar la topología del bolsillo con la forma de los ligandos.

Los parámetros de entrada son los que se muestran en el formulario del protocolo (Figura 9). Éstos son:

- ID de la GPU que se quiere usar para el mapeado de la superficie de la proteína. En este campo se indica 0 si no se quiere utilizar ninguna GPU (el tiempo de cómputo aumenta) o 1, 2, 3, etc. si se quiere usar alguna de las GPU del ordenador, seleccionando una según su numero de referencia.

- Estructura tridimensional de la proteína, ya preparada, usada en el protocolo anterior.

- Residuo diana sobre el que se reconstruirá la forma del bolsillo de la proteína. Este residuo debe estar contenido en el bolsillo para que el mapeo de la superficie sea en torno a él. Además, se pueden mapear superficies de interacción más grandes de la proteína seleccionando residuos adicionales de la proteína que flanqueen dicha región.

- Booleano para cambiar el punto desde donde los "rayos" o vectores tienen su origen. Por defecto, para generar la forma del bolsillo, los vectores tienen su origen en un punto a 30 Å hacia el centro geométrico de la estructura. Sin embargo, esto puede cambiarse y hacer que se lancen desde un residuo en concreto de la proteína, seleccionándolo en el campo "*Residue origin*".

- Booleano para incluir las cargas electroestáticas de la proteína, almacenadas en la grid tridimensional generada, en la confección de la forma del bolsillo. En caso de afirmativo, se debe introducir el objeto *gridAGD* que apunta al fichero de la *grid*.

- Booleano que indica la posibilidad de utilizar una semilla constante para el uso del programa de Rosetta, *make_ray_files.static.linuxccrelease* (para sistemas operativos Linux donde no se ha seleccionado el uso de GPU) o *make_ray_files.opencl.linuxccrealease* (cuando se ha seleccionado una GPU), que genera un fichero de texto y en formato *pdb*, con el mapeado de la superficie como se puede ver en la Figura 9 B. De esta forma, el mapeado sería constante y siempre igual. En caso afirmativo, se establece por defecto la semilla 1111111, pudiendo cambiarse.

Con estos parámetros de entrada, el protocolo, en primer lugar, va a ejecutar una función que lanza una línea de comando a la terminal de Linux para el programa *make_ray_files*. Este va a generae dos ficheros, uno en formato pdb y otro en txt con la forma del bolsillo, guardados en la carpeta destinada al protocolo, y que se usarán para construir la salida del protocolo, que dependiendo de si se incluyen las cargas electroestáticas o no, serán dos o tres objetos. Los comunes son dos objetos, *RaysStruct* y *RaysProtein*, que apuntan al fichero pdb y de texto mencionado anteriormente. En el caso de incluir las cargas, se genera un tercer objeto *GridAGD*, debido a que el programa *make_ray_files* produce una nueva grid electroestática que se usará en el protocolo de Rosetta DARC.

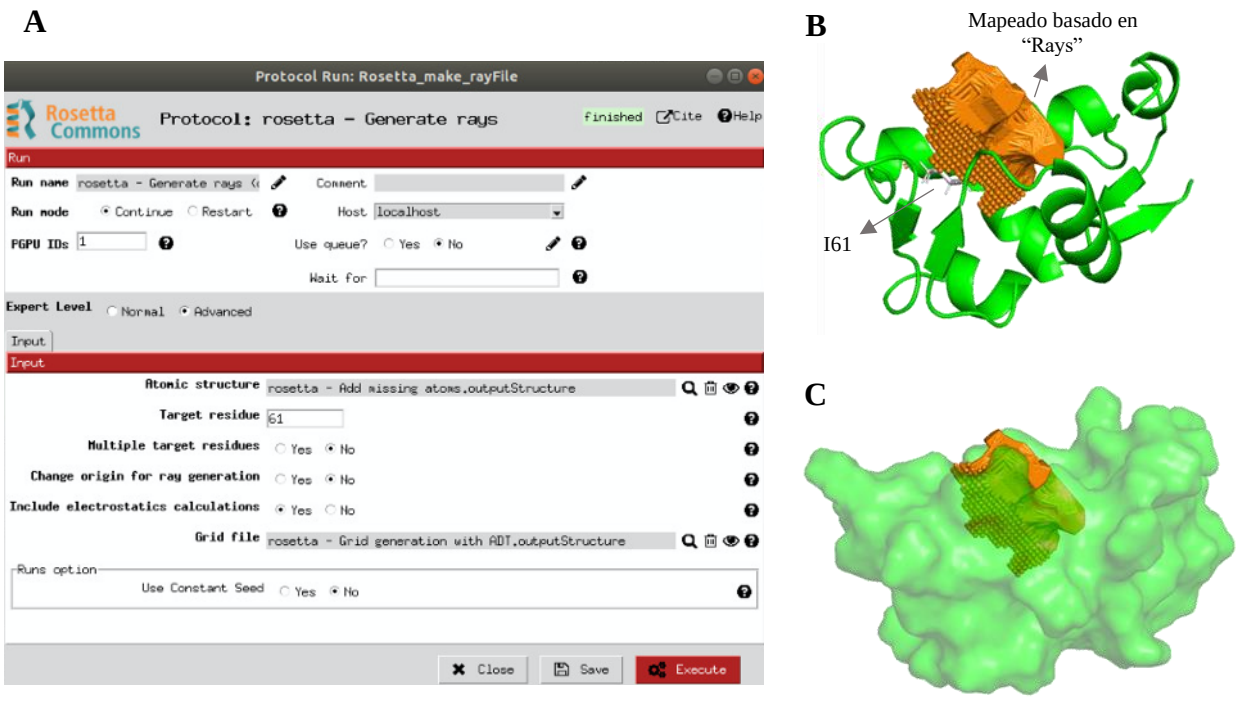


***Figura* 9. A)** Formulario del protocolo *Generate rays* con los parámetros de entrada necesarios para mapear la superficie de la proteína (PDB: 4ERF cadena A, obtenida previamente) en torno al residuo diana seleccionado, que en este caso es la isoleucina 61. **B)** Representación de la estructura secundaria de la cadena proteica utilizada (verde) y el mapeado resultante del protocolo realizado de la superficie, con una concavidad hacia el interior de la proteína, donde estaría presente la isoleucina 61, aminoácido apolar, que contribuiría a la característica hidrofóbica de los bolsillos de PPI. **C)** Superficie de la proteína MDM2, donde se observa el mapeado sobre la superficie de la proteína, respetando la topología de esta.

3.3.2.4.4. DARC Ligand Preparation

El propósito de este protocolo es realizar la preparación del conjunto de moléculas que se quieren utilizar en el *docking* mediante la asignación de las cargas parciales de los distintos átomos que forman la molécula y el ajuste del estado de la molécula para un pH dado, generalmente el pH fisiológico de 7,4. Además, como Rosetta DARC lo necesita en su flujo de trabajo, este protocolo permite la generación de confórmeros para cada molécula, rotando los enlaces covalentes simples de la molécula. La asignación de cargas y generación de confórmeros se realiza gracias a funciones integradas en el paquete OpenBabel [49], instalado junto al *plugin*.

Para este propósito, los parámetros de entrada (Figura 10) son:

- Un objeto de tipo *SetOfSmallMolecules*, que guarda la ruta para cada archivo de las moléculas que se quieren probar durante el cribado virtual con Rosetta DARC u otro programa para *docking*, ya que este tipo de objeto simplemente almacena estas moléculas.

- Método para asignar las cargas a los átomos de las moléculas. Por defecto el método que se usa es el método de *Marsilli-Gasteiger*, debido a que las cargas de la proteína se asignan en la generación de la *grid* usando este método, manteniendo así la concordancia en la asignación (estas cargas no variarán durante la generación de los confórmeros debido a que se basan en la electronegatividad de los átomos). Sin embargo, en el protocolo se pueden elegir distintos métodos de asignación de cargas, más avanzados que el de *Marsilli-Gasteiger*. Estos son: *mmff94* (basada en los campos de fuerza y es especialmente útil cuando se quieren estudiar interacciones intramoleculares), *eem*, *qeq*, *eq*eq y *qtpie*, basados los 4 en la asignación de carga en base a la electronegatividad, transferencia de carga y polarización.

- Booleano para indicar si se quiere establecer un pH, variando los hidrógenos de la molécula de acuerdo con este. Por defecto se usará el pH fisiológico. En este punto, aun no se ha incluido un método para establecer los estados de protonación de la proteína, como PropKa3 [51], siendo esto un trabajo futuro para el desarrollo de una plataforma completa de cribado virtual, por lo que de forma general aun no se cambiará el pH de los ligandos.

- Booleano para indicar si se desea o no generar confórmeros de cada molécula. En caso de usarlo para el cribado virtual con Rosetta DARC, se deben generar estos confórmeros, ya que asegura una mayor exploración del espacio conformacional durante el *docking*, disminuyendo la tasa de falsos positivos y negativos.

- Método de generación de confórmeros. OpenBabel tiene dos métodos para la producción de confórmeros a partir de una única molécula: un método basado en algoritmos genéticos y otro método denominado ConFab [52]. Ambos métodos aseguran el criterio de diversidad en cuanto a las posibles estructuras de los confórmeros, El método más adecuado por optimizar un mínimo de energía es el algoritmo genético, siendo el que se ha escogido por defecto.

  El algoritmo genético genera el número de confórmeros indicado a través de la producción de una serie de confórmeros que, tras una serie de iteraciones se seleccionan en base a la mayor diversidad de RMSD (del inglés, *root-mean-square deviation*) de las posiciones atómicas, pero que minimiza la energía de las moléculas.

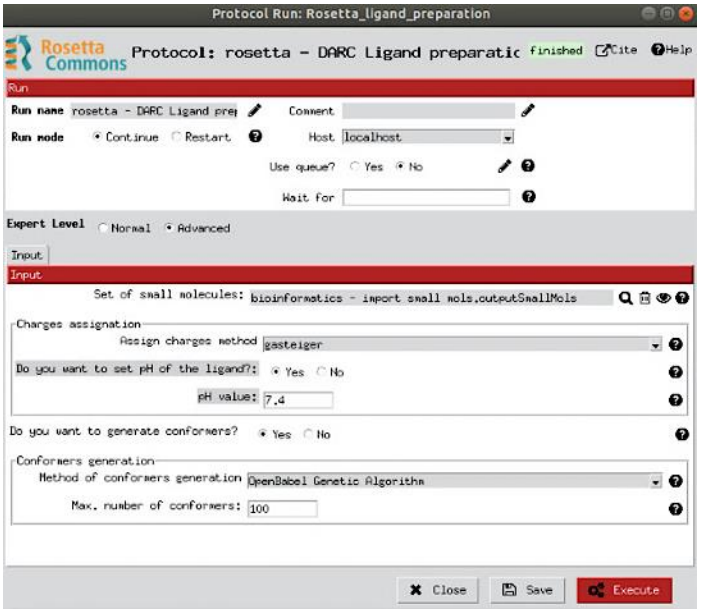
Confab, a diferencia del algoritmo genético, es un método sistemático que genera todos confórmeros de baja energía para la molécula, ordenándolos de acuerdo con el RMSD y la energía y se elige el número indicado mediante el establecimiento de dos límites de RMSD y energía, que por defecto es 0,5 Å y 50 Kcal/mol, respectivamente

- Número máximo de confórmeros a generar. Por defecto este número es de 200, ya que es un número suficiente para moléculas con un elevado número de grados de libertad.

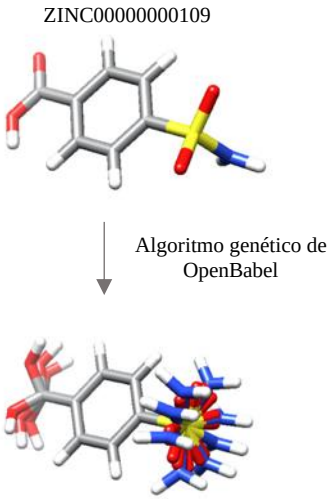
Con estos parámetros de entrada y usando OpenBabel, en primer lugar, se asignan las cargas según el método seleccionado. Para esta asignación se convierten los ficheros a formato sdf, ya que así se recomienda en el tutorial de Rosetta DARC [53]. Con los ficheros en el formato adecuado se asignan las cargas y se añaden o eliminan los hidrógenos de los grupos donantes o aceptores de hidrógenos de acuerdo con el pH seleccionado. Posteriormente, se generan los confórmeros, que se guardan en archivos con múltiples mol2. A continuación, para ejecutar Rosetta DARC, se necesita de un fichero de parámetros, en formato *params*, que describe a cada molécula, sus tipos de átomos, enlaces y coordenadas, para que el programa pueda usarlo. Este fichero se crea usando un script de Python que proporciona Rosetta, *batch_molfile_to params.py*.

Por último, habiendo guardado para cada molécula el archivo de confórmeros, parámetros y el generado en formato mol2, tras la adición de cargas, se genera un objeto *SetOfSmallMolecules*, que guarda las rutas a esos ficheros para cada compuesto procesado. Esta es la salida de este protocolo y que se usará en el último protocolo donde se realiza el *docking* con Rosetta DARC

A



B



C

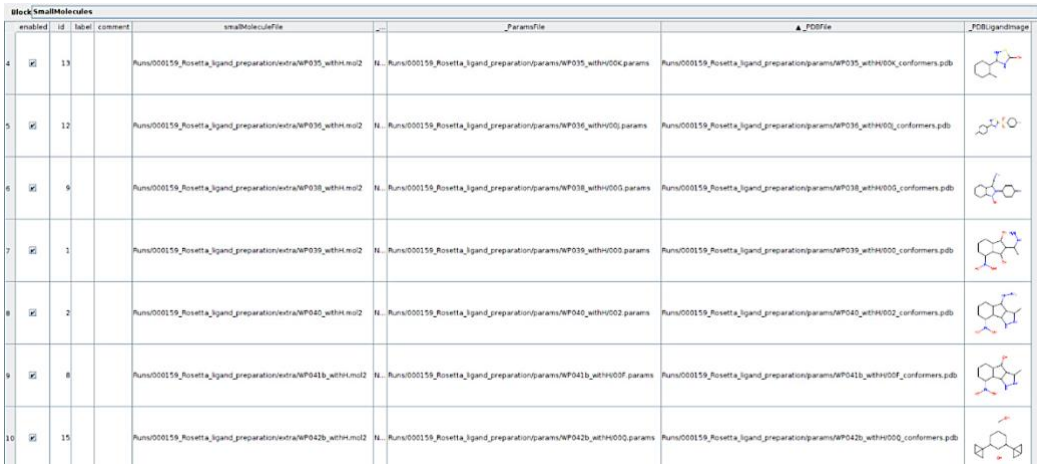


***Figura* 10. A)** Formulario del protocolo *DARC Ligand Preparation* con los parámetros de entrada necesarios para preparar el conjunto de compuestos importados para realizar el *docking*. **B)** Representación de una de las moléculas introducidas a este protocolo en formato mol2 (ZINC109, Butamisola o *2-metil-N-[3-(2,3,5,6-tetrahidroimidazo[2,1-b][1,3]tiazol-6-il)fenil]propanamida*). Tras generar los confórmeros de baja energía de esta molécula, con el algoritmo genético de OpenBabel, se obtiene un conjunto de ellos representados juntos, donde se aprecia la rotación de enlaces simples en cada molécula superpuesta. C) Objeto de salida del protocolo, *SetOfSmallMolecules*, donde para cada compuesto introducido genera un fichero con los distintos confórmeros en formato multi-mol2, un fichero de parámetros y un fichero pdb con dichos confórmeros, además, de la molécula preparada en formato mol2 también (columna *smallMoleculeFile*)

3.3.2.4.5. DARC

Este protocolo realizará el docking de cada uno de los compuestos y sus confórmeros sobre la proteína diana que ya ha sido preparada. Para ello, los parámetros de entrada necesarios son (Figura 11):

- ID de la GPU a utilizar durante el docking, debido a que es un proceso computacionalmente costoso para una CPU.

- Objeto *AtomStruct* con la estructura tridimensional de la proteína, ya preparada.

- Objeto *SetOfSmallMolecules* generado durante el protocolo de preparación de ligandos, que contiene la dirección a los ficheros de los ligandos (ligando original, sus confórmeros y el fichero de parámetros).

- Objeto *RaysProtein* que contiene el mapeado de la superficie de la proteína sobre la que se quiere hacer el *docking*

- Booleano para indicar si el *docking* se realiza solo en base a la complementariedad de forma, usando para ello el primer sumando de la función de puntuación (Fórmula (1)) o se tiene en cuenta también la complementariedad de carga, añadiendo el segundo sumando a la Fórmula (1). En caso negativo, será necesario añadir al formulario el objeto *gridAGD* que contiene la grid electroestática de la proteína generada previamente.

- Booleano para usar la búsqueda y evaluación de los confórmeros "*on the fly*". Esta característica acelera la evaluación de cada uno de los confórmeros en el mismo proceso de *docking*, en vez de hacer un docking secuencial para cada uno de dichos confórmeros. En el proceso secuencial, se le da a cada confórmero la misma cantidad o peso en el muestreo (número de partículas), mientras que en la aproximación "*on the fly*", los confórmeros se disponen en el espacio conformacional a evaluar (con un único número de partículas por molécula y no por confórmero), siendo ahora este mayor. En este caso se recomienda que el número de iteraciones y partículas utilizadas en la PSO sea mayor, para alcanzar la solución optima.

- Booleano para indicar si se minimizan los complejos logando-proteína durante el proceso de docking usando, además, la función de puntuación *ref2015*. En caso afirmativo, Rosetta DARC proporcionará más métricas acerca de la interacción proteína-ligando y la puntuación global tendrá en cuenta más parámetros. Si bien es verdad que la evaluación es más exhaustiva, el tiempo de ejecución también es mayor.

- Los parámetros avanzados se refieren a aquellos relacionados con la ejecución del algoritmo PSO (número de iteraciones y partículas), que cuanto mayor sean 1) más gasto computacional tiene el proceso y 2) mayor es la probabilidad de alcanzar el mínimo global en la generación del confórmero que mejor se una al bolsillo; y con los pesos considerados en la función de puntuación, para aquellos puntos que no toquen el bolsillo, el ligando o puntos del ligando que se alejan del bolsillo mapeado.

- Booleano que indica la posibilidad de utilizar una semilla constante para el uso del programa de Rosetta. En caso afirmativo, se establece por defecto la semilla 1111111.

Con estos parámetros de entrada, se ejecutará Rosetta DARC para cada compuesto considerado en el *SetOfSmallMolecules*, calculando la puntuación correspondiente a la función de puntuación (Formula (1)), vista en el apartado donde se explica el funcionamiento de Rosetta DARC.

La salida de este protocolo es una tabla (objeto *SetScores*) donde se almacena el ID de la molécula y la puntuación obtenida de Rosetta DARC para la unión del compuesto y la proteína diana, pudiéndose ordenar estos compuestos por dicha puntuación. En el caso de que se realice la minimización, la función de puntuación *ref2015* reporta otros indicadores adicionales de la unión en el complejo proteína-ligando, como:

- *Total Energy* (Kcal/mol): Energía libre total del complejo proteína-ligando.
- *Interface Energy* (Kcal/mol): Energía implicada en la unión entre la superficie de la proteína y el ligando, calculada como la suma de las energías de los pares de uniones posibles entre cada residuo de la superficie de la proteína y el ligando. Si estos no interaccionan, la energía libre involucrada es de 0 Kcal/mol. Cuanto más negativo sea, más estable será el complejo.
- *Interface HB*. Número de puentes de hidrógeno que se predicen que se producirán entre los residuos de la superficie de la proteína y el ligando unido mediante *docking*, siendo una métrica que refleja la estabilidad de la unión en el complejo.
- *Interface_Unsat*. Número de grupos polares que se encuentran en la interfaz sin formar puentes de hidrógeno compensatorios, y que, por lo tanto, son desfavorables desde el punto de vista de la solvatación.
- *Theta Lig*: Fracción del compuesto que está expuesto al solvente en el complejo.

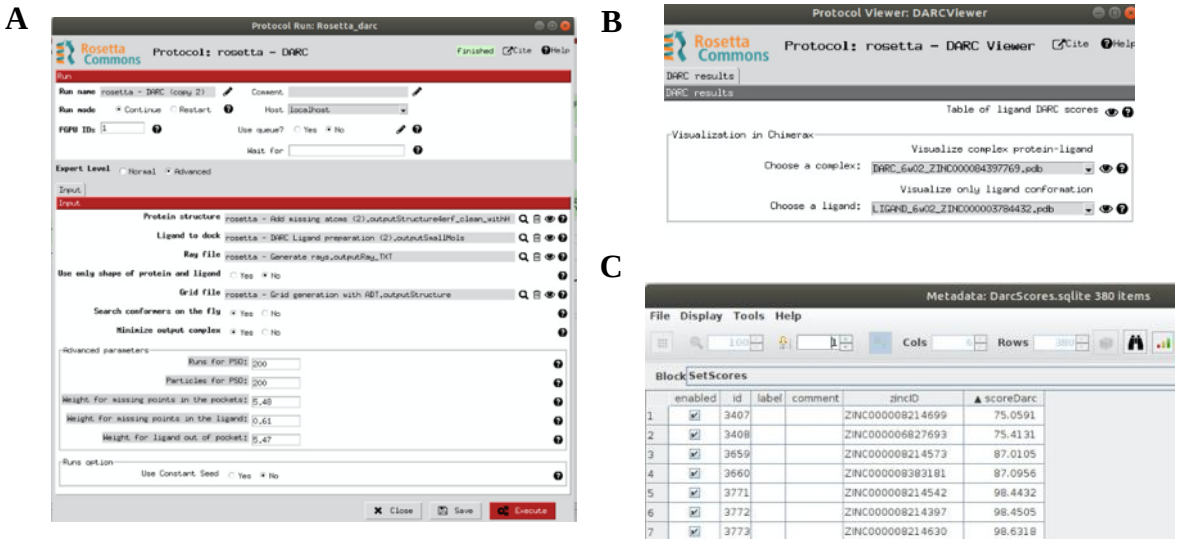


***Figura* 11. A)** Formulario del protocolo *DARC*, con los parámetros de entrada necesarios para llevar a cabo un proceso de *docking* único o cribado virtual, dependiendo del número de moléculas introducidas en el *SetOf SmallMolecules*. Los parámetros mostrados en gris son parámetros avanzados que el usuario puede modificar pero que presentan valores por defecto, aunque haya algunos muy relevantes para llegar a un docking óptimo como el número de iteraciones y partículas de PSO, que debe ser mayor, si se utiliza la opción de búsqueda de confórmeros "*on the fly*". **B)** Visor donde se pueden visualizar los resultados del docking en una tabla y los distintos complejos proteína-ligando generados. **C)** Tabla con las puntuaciones del *docking* para cada molécula.

3.3.2.4.6. Convert formats

Este protocolo tiene como objetivo la conversión entre formatos de estructura de proteínas y formatos de estructuras de moléculas pequeñas. De hecho, este protocolo fue creado para usarlo especialmente para los conjuntos de moléculas utilizadas en docking ya que, en el caso concreto de Rosetta DARC, solo admite dos tipos de formatos de estructura tridimensional de ligandos, *pdb* y *mol2*, y desde las bases de datos como ZINC15 se pueden descargar en múltiples formatos 2D y 3D. Para la conversión entre formatos se utiliza las funciones proporcionadas por el paquete OpenBabel.

Los parámetros de entrada (Figura 12) son un conjunto de moléculas pequeñas (objeto: *SetOfSmallMolecules*) o una estructura como objeto *AtomStruct*, al igual que los de salida dependiendo de dicha entra. A partir de la entrada, se obtiene el formato de los ficheros de cada molécula o estructura, permitiendo que, para un conjunto de moléculas con diversos formatos, se pueda obtener un conjunto con los archivos en un solo formato, elegido en estos parámetros de entrada como formato de salida. Los formatos disponibles de salida para los conjuntos de moléculas son: pdb, cif, mol2, sdf, smiles o smi. Para las estructuras de proteínas son: pdb y cif.

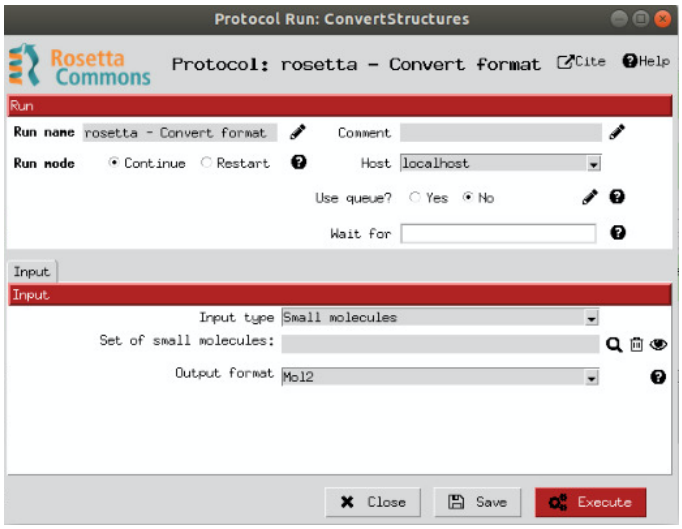


***Figura* 12.** Formulario del protocolo *Convert Formats* con los parámetros de entrada necesarios para convertir el formato de los archivos de estructura macromoleculares proteicas (pdb, cif) y moléculas pequeñas (pdb, cif, mol2, sdf, smi).

3.3.2.5. Tests

Esta carpeta incluye los *tests* de integración en *Scipion* de cada uno de los protocolos. Cada protocolo cuenta con un test que asegura que funcione el protocolo creado en *Scipion* y se integre con los demás protocolos que le preceden, ya que se emula la generación de un proyecto con datos de entrada conocidos y cuya salida se puede evaluar con precisión.

4. Casos de uso

Para probar la utilidad del programa de *docking* Rosetta DARC y su integración en *Scipion* se han elaborado dos casos de uso distintos y de interés biológico y social. Se probará el *software* frente a los dos tipos de bolsillos clasificados en la introducción y presentes en las proteínas: bolsillo tradicional hidrofílico y bolsillo superficial (especialmente implicado en las PPI).

4.1. Reposicionamiento de fármacos. Cribado virtual sobre la el macrodominio de la proteína no estructural 3 (NSP3) del SARS-CoV-2

Este caso de uso está basado en un artículo publicado en la revista *Science Advances* en abril de 2021, sobre el cribado virtual y cristalográfico de fragmentos moleculares en el Mac1 de la NSP3 del SARS-CoV-2 [54].

4.1.1 Introducción

A junio de 2021, el síndrome respiratorio agudo severo, producido por el coronavirus 2, (SARS-CoV-2) ha afectado a más de 175 millones de personas, causando más de 3,8 millones de muertes en el mundo [55]. La búsqueda de terapias antivirales efectivas para tratar el SARS-CoV-2 ha sido amplia y variada, desde el reposicionamiento de fármacos hasta el descubrimiento *de novo* de medicamentos [56].

Una de las dianas terapéuticas, debido a su participación en la inhibición de la inmunidad innata del huésped, y, que, por consecuencia, contribuye a la virulencia del virus, es el macrodominio (Mac1) de la proteína no estructural 3 (NSP3), proteína con más de 10 dominios y 2000 residuos. Para ilustrar la importancia de bloquear este macrodominio, es necesario conocer cuál es su función [54].

De esta forma, cuando el SARS-CoV-2 forma un complejo con la enzima convertidora de angiotensina 2 (ACE2) y entra en las células, desencadenando la activación del sistema renina-angiotensina. La renina transforma el angiotensinógeno en angiotensina I (Ang-I), que, a su vez, por la enzima convertidora de angiotensina, se procesa a angiotensina II (Ang-II). Debido al aumento de Ang-II y la reducción del ACE2 libre, el receptor de angiotensina tipo I (AT1R) se activa y desencadena una cascada de señalización a través de la activación de NADPH oxidasas, que induce un intenso estrés oxidativo. Los ROS (radicales libres de oxígenos) provocan la rotura de las cadenas de ADN, lo que activa a la polimerasa poli ADP-ribosa (PARP) para intentar reparar el daño en el ADN. Las PARP se modifican transfiriendo ADP-ribosa, procedente del NADH, a sí misma y a otras proteínas que se encuentren en la célula, tanto virales como propias, llevando a la activación del interferón-γ y distintas citoquinas pro-inflamatorias y antivirales, así como a la inhibición de la traducción de proteínas virales y de la reproducción del virus [57].

El Mac1 de NSP3, con su actividad ADP-ribosil (ADPr) hidrolasa, contrarresta esta señalización antiviral a través de la hidrólisis de la ADP-ribosa unida a las proteínas del virus. Al inhibir esta actividad, se podría generar una mayor y eficaz respuesta frente al virus. Solo existe un inhibidor de la actividad ADPr hidrolasa bien caracterizado con una farmacología y selectividad adecuadas frente a la hidrolasa específica, PDD00017273. Al ampliar el espacio químico y buscar inhibidores, en moléculas ya aprobadas por las agencias reguladoras del medicamento, mediante cribado virtual, se

podría generar una terapia rápida y accesible y dar puntos de partida para generar nuevos inhibidores de la hidrolasa [54].

En este caso de uso con Rosetta DARC se pretende hacer una aproximación en el descubrimiento de medicamentos mediante reposicionamiento de estos sobre el bolsillo hidrofílico de la hidrolasa, ya que la librería usada para buscar compuestos que tengan la suficiente complementariedad de forma y afinidad, es una que contiene ya fármacos aprobados.

4.1.2 Materiales y métodos

I. Librería de compuestos

La librería de moléculas se descargó de la base de datos ZINC15 (https://zinc15.docking.org/). Del conjunto total de compuestos de dicha base de datos (1.810.070.912 entradas) se seleccionaron aquellos que habían sido probados *in vivo* y en humanos, aprobados por la mayoría de las agencias reguladoras del medicamento del mundo (EMA (Europa), FDA (Estados Unidos), NIHS (Japón), TGA (Australia), etc.), que se encuentran a la venta y se pueden conseguir comercialmente en un periodo muy corto de tiempo por su alta disponibilidad. De esta manera, se descargaron para testar 3838 compuestos, en formato mol2, que ya se encuentran en el mercado, la mayoría, como fármacos. Dentro de esta librería se incluyen aquellos 46 compuestos que se seleccionaron en el artículo de referencia y como, control positivo, se ha añadido, además, el ADPr, que se une al centro activo de la hidrolasa (ZINC000032786521).

II. Preparación de la librería de ligandos

Las moléculas se prepararon utilizando el protocolo *DARC Ligand Preparation* de *Scipion*. Durante esta preparación, previa al *docking*, se agregaron hidrógenos y se asignaron las cargas parciales a cada átomo según el método de *Marsilli-Gasteiger*. Finalmente, se generaron 100 confórmeros de cada ligando, utilizando el algoritmo genético de OpenBabel, y se generaron los archivos de parámetros necesarios para Rosetta DARC.

III. Proteína diana y preparación de la estructura

El *docking* se realizó contra la estructura cristalina del Mac1 de la proteína Nsp3 del SARS-CoV-2 unido a ADPr (PDB: 6W02), siendo la misma estructura que usaron en el artículo [54] y que usaremos para comparar los resultados obtenidos.

Para la preparación de la estructura se utilizó el protocolo *DARC Protein Preparation* de *Scipion.* Con este, se eliminaron las moléculas de aguas presentes en la estructura, se eliminó el ligando unido a cada una de las cadenas y se seleccionó la cadena A del archivo pdb. Además, se añadieron a la estructura los hidrógenos y átomos que no hubieran sido trazados en las cadenas laterales.

IV. Mapeado del bolsillo de unión en la proteína y generación de la rejilla electroestática

El bolsillo donde se encuentra el centro activo de la Nsp3 (y tiene lugar la hidrolisis del motivo ADP-ribosa) es básicamente superficial. Este bolsillo se compone de dos sub-sitios, el sub-sitio de la derecha (o proximal, donde se une la adenosina) o el de la izquierda (o distal, donde se une la ribosa).

Ambos sub-sitios están conectados por un pequeño puente estructural que ocultaría el enlace difosfato que une la adenosina con la ribosa (Figura 12).

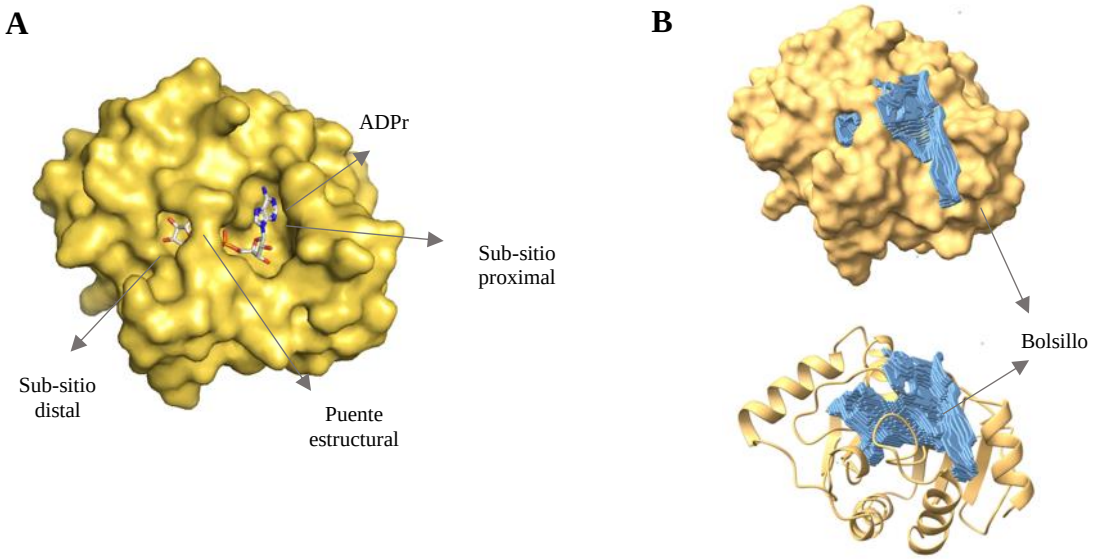


***Figura* 12. A)** Cadena A de la estructura cristalina del Mac1 de la proteína Nsp3 del SARS-CoV-2 unido a ADPr (PDB: 6W02), donde se muestran los distintos sub-sitios del bolsillo que presenta la actividad ADPr hidrolasa. **B)** En azul, sobre la estructura del Mac1 (amarillo), se representa el mapeado del bolsillo, con una zona proximal superficial donde se pretende hacer docking. Arriba se muestra la superficie de la proteína y abajo, una representación de la estructura secundaria de la misma.

Se construyó la *grid* electroestática sobre la proteína (con el protocolo *Grid generation with ADT*), siendo las dimensiones de está de 101 x 101 x 101 $Å^3$, ya que el radio de circunferencia que cubre la proteína es de 28 Å. Se seleccionó un espaciado entre cada punto de la rejilla de 0.375 Å. Así mismo, se asignaron cargas parciales a los átomos de la proteína usando el método de *Marsilli-Gasteiger* implementado en ADT.

A continuación, se definió la forma del bolsillo utilizando el protocolo *Generate rays* de *Scipion*, para lo que se seleccionó el residuo L126, localizado en el bolsillo proximal. También se definió el mismo bolsillo utilizando dos residuos, L126 y A38, que se encuentran en dicho bolsillo y en cada uno de los dos sub-sitios mencionados y los resultados obtenidos eran iguales.

V. Cribado virtual usando Rosetta DARC.

Después de preparar la librería de posibles ligandos y la proteína, se sometieron al protocolo de *docking* en *Scipion*, *DARC*. Para ello, en el algoritmo de búsqueda conformacional (PSO), se tuvieron en cuenta 200 partículas y 200 iteraciones para su ejecución, siendo el doble de lo recomendado ya que se utilizó búsqueda de confórmeros "*on the fly",* donde al probar en un mismo procedimiento de *docking* los confórmeros generados a la vez, se amplia el espacio conformacional a explorar y la confluencia al mínimo podría necesitar de mayor número de iteraciones. Así mismo, al tener un número elevado de compuestos que probar, no se realizó una minimización con el método *fullatom* de Rosetta al conjunto total.

Para realizar un análisis comparativo del método Rosetta DARC, se hicieron dos aproximaciones. En la primera de ellas, se hizo *docking* sólo teniendo en la complementariedad de forma entre el bolsillo mapeado y los ligados en la función de puntuación que utiliza Rosetta DARC (primer sumando de la Fórmula (1). En la segunda, se tuvo en cuenta tanto la complementariedad de carga como de forma, usando para ello la *grid* electroestática calculada previamente sobre la proteína.

Posteriormente, las moléculas se clasificaron según dicha función de puntuación (Tabla S1) y se obtuvo el 10 % (383 compuestos) con mejores resultados (menor valor en la función de puntuación, que siempre es un valor positivo), a los cuales se les aplicó la minimización con el método *fullatom* de Rosetta.

El cribado virtual se realizó usando una GPU *GeForce RTX 2060 (1200 MHz)* con 40 procesadores y 1024 hilos. El proceso completo de *docking* necesitó aproximadamente de 2 horas.

4.1.3 Resultados y discusión

I. Flujo de trabajo en *Scipion*

El flujo de trabajo realizado para este caso de uso de puede visualizar en la Figura 4, donde se llevó a cabo el cribado virtual utilizando cada uno de los pasos mencionados en Materiales y Métodos. En total, se utilizaron 11 protocolos, más aquellos correspondientes a la exportación de los resultados de puntuación para cada compuesto a archivos en formato csv.

De esta forma, se puede automatizar, esquematizar y trazar de manera sencilla y visual uno o varios experimentos de cribado virtual sobre una o, incluso varias, proteínas diana utilizando el mismo conjunto de ligandos (*docking* multi-diana). Al finalizar todos los protocolos esquematizados se podría realizar un análisis cuantitativo y estructural de los mejores resultados, ya minimizados, debido que Rosetta DARC, construye los complejos proteína-ligando y los almacena en formato pdb, que podemos obtener en el directorio de los protocolos donde se realiza el *docking* o en el visor generado.

II. Cribado virtual. Identificación de posibles inhibidores de la actividad hidrolasa

Los resultados del cribado virtual, donde se ha puntuado cada acoplamiento entre cada una de las 3839 moléculas probadas y el bolsillo del Mac1, que presenta la actividad ADPr hidrolasa, en base a la complementariedad de carga y forma se encuentran en la Tabla S1. En dicha tabla los compuestos están ordenandos según su puntuación, siendo aquellas con menor valor, las que mejor complementariedad presentan.

Entre el 10% de los mejores resultados (383 compuestos) podrían encontrarse moléculas que interactuasen con el bolsillo de la ADPr hidrolasa, bloqueando competitivamente la entrada de las ADPr unidas a proteínas, impidiendo que se hidrolicen estas modificaciones posttraduccionales y la célula pueda activar su mecanismo antiviral.

Las primeras moléculas con mejor puntuación (ZINC000008214699, ZINC000008214535, ZINC000008383181, etc.) son moléculas pequeñas, que apenas cubren y se unen al sub-sitio proximal del Mac1, por lo que es poco probable que inhiban al enzima y compitan con el sustrato original, el ADPr. Aun así, una de ellas, la 2-aminopteridin-4(3H)-ona (ZINC000017819771) (puesto 132 en el *ranking*), utilizada en la fabricación del metotrexato, fármaco antiinflamatorio usado en el tratamiento de la artritis y algunos tipos de cáncer [58], podría ser un candidato a probarse experimentalmente. Tanto el ADPr, ligando natural del Mac1 como el compuesto ZINC000017819771, con una estructura similar a la adenosina, se unen al sitio proximal del bolsillo y puente estructural, a través de distintos puentes de hidrógeno e interacciones polares entre los compuestos y las cadenas laterales mostradas en la Figura 13. De esta forma, este compuesto podría impedir la entrada y unión de la ADPr, bloqueando su hidrólisis.

Sin embargo, esto no deja de ser una predicción y un punto de partida para generar nuevas moléculas con mejor acoplamiento y analizar, en primer lugar, la dinámica de interacción entre los compuestos con mejor puntuación y, posteriormente, las propiedades fisicoquímicas y biológicas (incluidas las propiedades ADMET, que al ser fármacos ya deben estar optimizadas), en cuanto al efecto y la selectividad de la unión en ensayos experimentales *in vitro*.

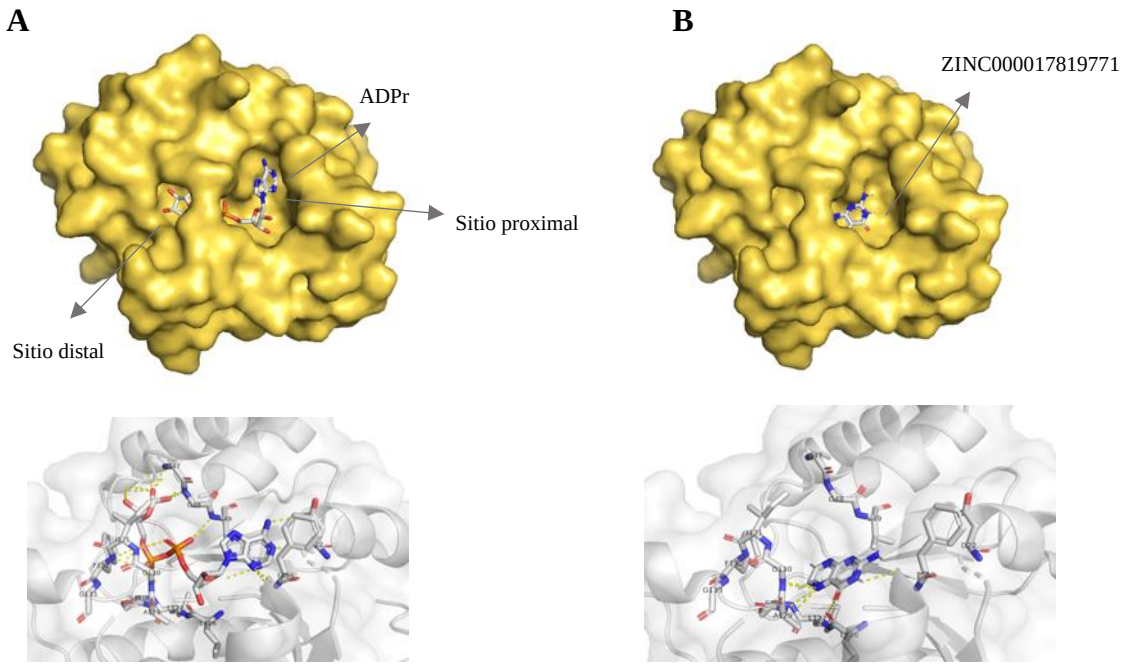


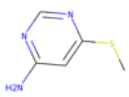
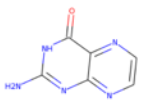
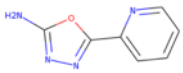
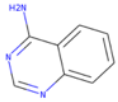
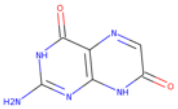
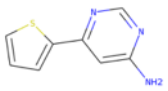
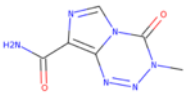
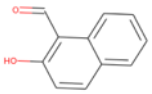
***Figura* 13. A) Arriba:** Cadena A de la estructura cristalina del Mac1 de la proteína Nsp3 del SARS-CoV-2 unido a ADPr (PDB: 6W02). **Abajo**: Residuos del bolsillo de Mac1 que interaccionan con el ADPr a través de puentes de hidrógenos, pintadas como rayas discontinuas amarillas. **B) Arriba:** Cadena A de la estructura cristalina del Mac1 de la proteína Nsp3 del SARS-CoV-2 unido al compuesto ZINC000017819771 (complejo obtenido tras el *docking*). **Abajo**: Residuos del bolsillo de Mac1 que interaccionan con el ZINC000017819771 a través de un número de puentes de hidrógenos mejor al ADPr con L127, A129 y G130.

III. Comparación con los resultados del artículo donde se identifican 46 compuestos candidatos.

Para testar la eficacia del método de *docking* integrado, Rosetta DARC, se ha realizado una comparación con los compuestos más prometedores obtenidos del estudio [54], donde a partir de la librería completa de ZINC15, mediante un cribado virtual y posterior análisis cristalográfico de los compuestos con mejor puntuación, obtuvieron un conjunto de moléculas que se unían al Mac1. En concreto 20 de ellos, mostraron una mayor capacidad de unión que el resto.

En la comparación se obtuvo que en el 10 % de los mejores resultados de Rosetta DARC habían sido identificados, como posibles ligandos de unión, 13 de los 46 compuestos que en este estudio [54] fueron postulados como *hits* (Figura 14 A). En el caso del 20 % de los mejores resultados, fueron identificados 25 de los 46 compuestos (Figura 14 B). En la Tabla 2 se muestra aquellos compuestos que fueron identificados en el 10 % como *hits* y su posición ordenada por puntuación entre los demás compuestos. Se observa como es común entre los compuestos que tienden a interaccionar con el bolsillo la estructura de anillos insaturados, similares a la adenina que compone el ADPr. Esto podría dar la clave para la generación de un fármaco más optimizado y selectivo, con una estructura similar a dicha adenina.

Tabla 2. Compuestos propuestos para la unión en el centro activo de la ADPr hidrolasa

ID	Nº ranking	Puntuación	Estructura
ZINC000001675805	108	191,867	
ZINC000017819771	131	203,506	
ZINC000002383176	165	215,184	
ZINC000000331945	166	215,2	
ZINC000014419577	167	215,202	
ZINC000044119054	198	226,779	
ZINC000001482184	243	238,428	
ZINC000000157162	279	250,173	

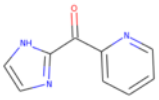
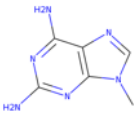
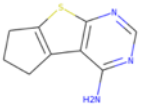
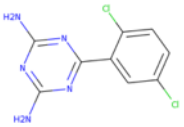
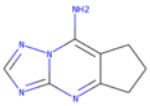
ZINC000004243442	281	250,176	
ZINC000026180281	288	250,232	
ZINC000000110227	324	261,789	
ZINC000000002645	365	273,354	
ZINC000007636250	381	273,582	

Tabla 2. Compuestos seleccionados como candidatos a unirse al Mac1 en el estudio [54], comparado con el puesto obtenido en nuestro cribado virtual con Rosetta DARC. Se indica para cada compuesto la puntuación obtenida y la estructura de cada uno.

Este hallazgo confirma que, en el caso del uso de un único programa de *docking*, con su función particular de puntuación, en este caso basada en el conocimiento, existe un gran número de falsos positivos entre el 10 o 20 % de mejores compuestos puntuados. Si bien es verdad que este porcentaje está enriquecido en compuestos "*hits*" con capacidad de unirse al bolsillo, proporcionando una reducción del número de compuestos a estudiar *in vitro* (y una disminución del tiempo y coste económico), no es el esperado, ya que debería ser un porcentaje mayor de acierto. Decimos que el conjunto obtenido tras nuestro cribado virtual está enriquecido en *hits* ya que presenta 2,8 veces más *hits* que lo que se esperaría al azar (para 10000 test aleatorios de intersección entre conjuntos, se obtuvo que la intersección media sería de 4,583 compuestos, con una desviación estándar de la media de 2,001). Aun así, el resultado no es el optimo ni el esperado.

Para solventar y enriquecer este porcentaje en compuestos capaces de unirse al bolsillo definido, se podría hacerse uso de combinación de los resultados de distintos programas de acoplamiento y funciones de puntuación, lo que se denomina *docking* consenso. Además, las condiciones y parámetros elegidos durante el acoplamiento y preparación de la librería de ligandos también pueden contribuir a un mejor o peor rendimiento en el *docking*. Este hecho se refleja en la Figura 14 C, donde se muestra que no se encuentran *hits* hasta el conjunto formado por el 70% de los compuestos mejor puntuados cuando se realiza el cribado virtual usando exclusivamente la complementariedad de forma. Esta gran diferencia indica que una buena elección de los parámetros en la función de puntuación es crucial para la correcta identificación de compuestos candidatos a fármacos. Así mismo, en el algoritmo PSO es muy relevante ajustar el número de iteraciones y partículas usadas, asegurando que se alcanza una conformación que minimice completamente la función de puntuación.

Por último, otra evidencia que refuerza, primero la optimización de los parámetros usados durante el *docking*, y segundo la necesidad de un análisis posterior y la inclusión de consensos de procesos de cribado virtual es el caso del ADPr, control positivo en la unión, que se encuentra en la posición 3299

del *ranking* del total de compuestos probados con una puntuación de 729,295. Esta mala puntuación se debe a que Rosetta DARC no ha conseguido ubicar de manera correcta el ligando en el interior del bolsillo definido y la orientación del ligando es contraria a la observada en las estructuras cristalinas disponibles (PDB: 6W02). Esto puede deberse a que, para esta molécula y el número de confórmeros construidos, 1) el número de iteraciones y partículas usadas en el PSO no hayan sido suficientes y no se haya realizado una exploración completa del espacio conformacional y 2) el número de combinaciones y orientaciones de la molécula, a la hora de construir los confórmeros, sobre la proteína no han sido suficientes ni adecuados, resultando en un acoplamiento sub-óptimo. De esta forma, se debería haber aumentado al doble el número de iteraciones y partículas y asegurar que los confórmeros presentan una diversidad adecuada de orientaciones en el espacio.

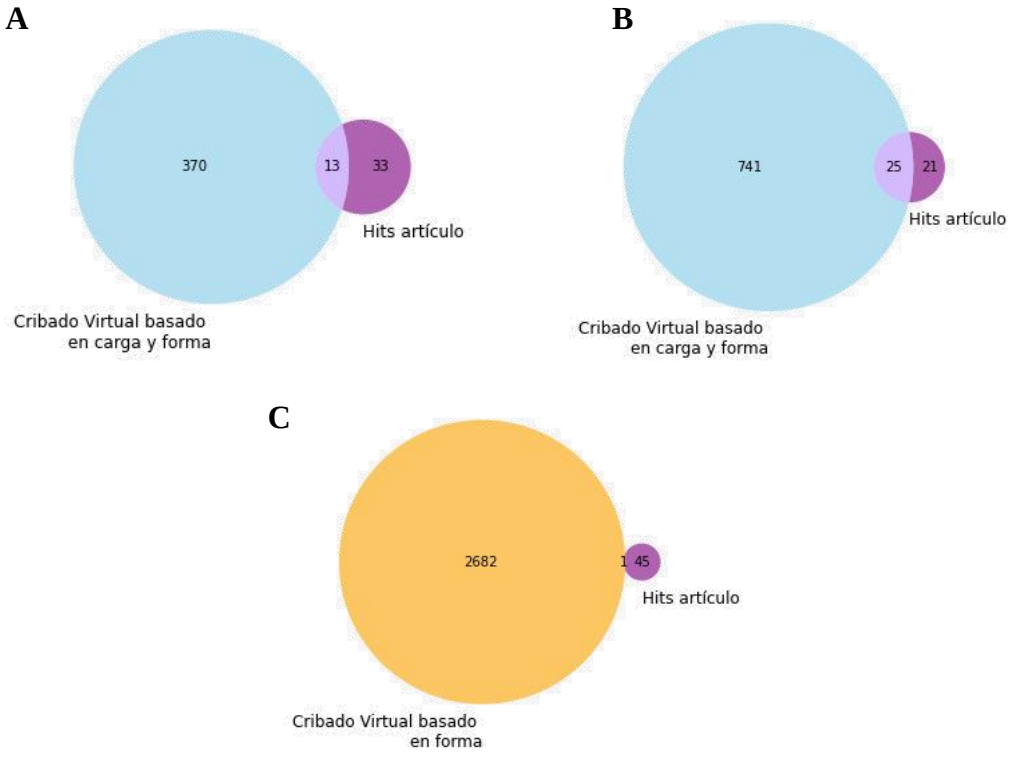


Figura 14. Diagramas de Venn donde se comparan los resultados del cribado virtual de este trabajo y los obtenidos en el artículo [54]. **A)** Comparación entre el 10 % de los mejores compuestos obtenidos en base a la complementariedad de forma y carga con los mejores compuestos encontrados en el artículo. **B)** Comparación entre el 20 % de los mejores compuestos obtenidos en base a la complementariedad de forma y carga con los mejores compuestos encontrados en el artículo. **C)** Comparación entre el 70 % de los mejores compuestos obtenidos en base a la complementariedad exclusivamente de forma con los mejores compuestos encontrados en el artículo.

IV. Comparación entre la puntuación basada en la complementariedad de carga y forma o solo basada en la forma.

Junto a los resultados del cribado virtual, donde se tuvo en cuenta la complementariedad de forma y carga entre el ligando y la proteína (Tabla S1), también se obtuvieron los resultados del cribado, donde solo se puntuó la complementariedad de forma (Tabla S2).

En los resultados del cribado virtual para el caso en el que se ha tenido en cuenta tanto la forma como la carga, se observa como el rango de puntuaciones oscila entre 75,0591 y 1407,4399, donde existe una mayor acumulación de compuestos en torno puntuaciones de 400 y 600 (Figura 15 A y B). En el caso del 10 % de los mejores compuestos, estas puntuaciones son menores a 273,68. En cambio, cuando se observan los resultados del *docking*, teniendo exclusivamente en cuenta la complementariedad de forma, el rango de puntuaciones abarca desde 3,2367 y 5,2137, ya que no existe en la función de puntuación un segundo sumando que penalice la falta de complementariedad de carga (Figura 15 C y D).

Comparando las dos aproximaciones, se observa como del 10 % de los mejores resultados de ambas, no coincide ningún compuesto (Figura 15 E). Tampoco en el 20 % mejor. No es hasta que se compara el 28% de los compuestos con mejor puntuación presentan (Figura 15 F), que coincide uno de ellos. Este compuesto (ZINC000004217580) ocupa la posición 897 (puntuación de 354,99) para la aproximación de forma y carga y la posición 1045 (puntuación 3,9621) en la aproximación de exclusivamente complementariedad de forma.

Sin entrar en detalles estructurales, estos resultados distintos podrían indicar que la complementariedad de carga, y no solo de forma, es muy relevante a la hora de encontrar compuestos que se unan al bolsillo e inhiban la unión, en este caso, del sustrato al enzima. En este bolsillo, las interacciones hidrofílicas predominan sobre las hidrofóbicas, debido a la naturaleza estructural del bolsillo y la naturaleza de la reacción de hidrólisis que cataliza. Por lo tanto, cuanto mayor sea el número de interacciones favorables entre un ligando y el sitio en el enzima y, exista una mejor complementariedad topológica, más probable es que este compuesto inhiba la actividad hidrolasa de Mac1.

Basándonos en estos resultados comparativos, los resultados procedentes del cribado virtual, utilizando carga y forma, que son los que muestran mayor tasa de acuerdo y acierto con el estudio previo [54], crean un punto de partida y una hoja de ruta para el desarrollo de inhibidores contra Mac1, que puedan ayudar a combatir la patogenicidad y virulencia del SARS-CoV-2.

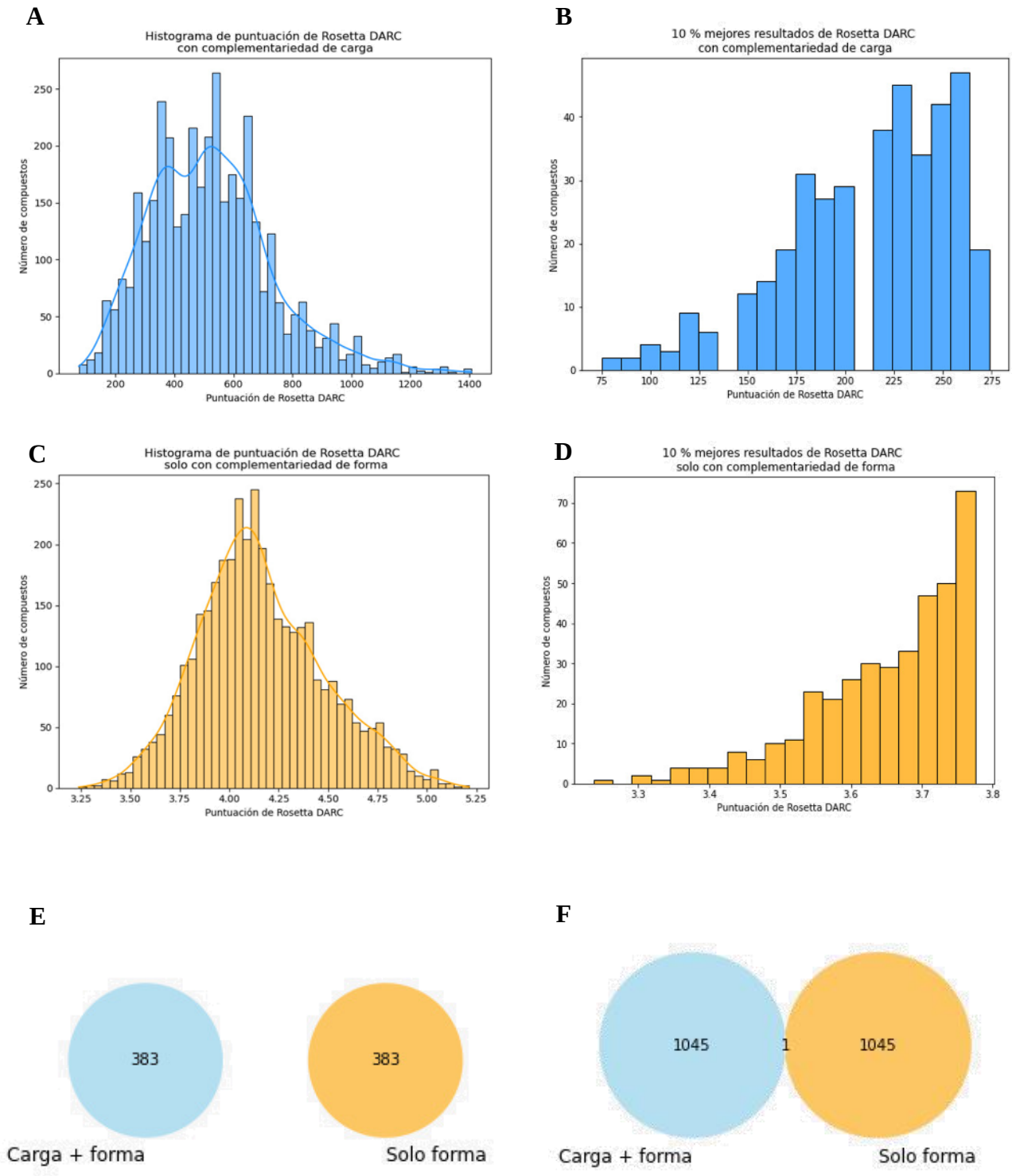


Figura 14. **A)** Histograma y curva de densidad de las puntuaciones de Rosetta DARC dada a cada compuesto en base a la complementariedad de carga y forma. **B)** Histograma donde se muestran solo el 10 % de los mejores resultados del histograma A. **C)** Histograma y curva de densidad de las puntuaciones de Rosetta DARC dada a cada compuesto en base a la complementariedad de forma. **D)** Histograma donde se muestran solo el 10 % de los mejores resultados del histograma C. **E y F)** Diagramas de Venn donde se comparan los mejores 10 % y 28 % resultados del cribado virtual obtenidos en base a la complementariedad de forma y carga (azul) con aquellos resultados obtenidos utilizando solo la complementariedad de forma (naranja).

4.2. Cribado virtual de compuestos moduladores de la interacción proteína-proteína Nsc-1/Ric8a implicada en el síndrome X frágil

4.2.1 Introducción

El síndrome X frágil (SXF) es un trastorno neurológico hereditario que causa discapacidad intelectual y autismo y que afecta aproximadamente a 1 de cada 2500 a 5000 hombres y a 1 de cada 4000 a 6000 mujeres y aun no tiene un tratamiento farmacológico eficaz. La mutación causante de la mayoría de los casos conocidos de SXF es una inserción de la secuencia CGG en el 5′ UTR del gen del retraso mental X frágil (*fmr1*), lo que resulta en la pérdida de la proteína del retraso mental X frágil (FMRP). FMRP es una proteína de unión a ARN que regula el transporte y la traducción de ARNm, que codifica gran parte del proteoma sináptico, el cual se vería alterado y se produciría una desviación del equilibrio sináptico que define la normalidad. Tanto el exceso como el déficit en el número de sinapsis pueden conducir a una patología como la del síndrome X frágil, trastornos del espectro autista (TEA) y esquizofrenia [59].

En el caso del SXF, las neuronas presentan espinas postsinápticas inmaduras (con morfología de cuello largo) y con mayor densidad. El aumento de sinapsis está relacionado con la disminución de FMRP. Sabiendo esto, y en base al conocimiento actual, diseñar terapias dirigidas a la disminución del número de sinapsis parece una buena estrategia. Sin embargo, la función neuronal requiere un control de la probabilidad de liberación de neurotransmisores por sinapsis, además del control del número de sinapsis, estando ambas propiedades co-reguladas de manera antagónica. Un enfoque potencialmente eficaz debería apuntar al mecanismo de señalización de esta co-regulación, en el que está implicado el sensor de calcio neuronal 1 (NCS-1), que interactúa con la proteína del factor de intercambio de guanina Ric8a para activar a las proteínas Gα y co-regular el número y la actividad de sinapsis [59].

El complejo NCS-1/Ric8a regula el del número de sinapsis y la probabilidad de liberación, por lo que la búsqueda de compuestos que pudieran unirse a la interfaz de ambas proteínas podría inhibir la formación del complejo y reducir el número de sinapsis y los efectos de un mayor número de estas [59].

En el estudio [59] se identificaron distintos compuestos (FD44 y SC16), que consiguieron inhibir la interacción de ambas proteínas y reducir el número de sinapsis hasta un estado normal y recuperar la capacidad de aprendizaje en un modelo de SXF de Drosophila.

Este ejemplo de PPI es adecuado para testar la cómo es de útil Rosetta DARC en este tipo de bolsillo, ya que fue diseñada principalmente para este objetivo.

4.2.2 Materiales y métodos

I. Librería de compuestos y su preparación

La librería química de compuestos usada es ha sido producida por el grupo de investigación “Química Médica y Biológica traslacional” de Nuria Eugenia Campillo Martín, Carmen Gil Ayuso-Gontán y Ana Martínez Gil, perteneciente al Centro de Investigaciones Biológicos (CIB). Esta librería está

compuesta por un total de 1982 moléculas, siendo la mayoría compuestos heterocíclicos. Las moléculas se prepararon utilizando el protocolo *DARC Ligand Preparation* de *Scipion*. Durante esta preparación, previa al *docking*, se agregaron hidrógenos, se calculó su carga y se asignaron las cargas parciales a cada átomo según el método de *Marsilli-Gasteiger*. Por último, se generaron 50 confórmeros para cada compuesto, utilizando el algoritmo genético de OpenBabel, y se generaron los archivos de parámetros necesarios para Rosetta DARC, de los cuales, 80 de ellos no pudieron generarse por problemas internos en los ficheros sdf de la librería.

II. Proteína diana y preparación de la estructura

El docking se realizó contra la estructura cristalina de NCS-1 (PDB: 1G8I). Para la preparación de la estructura se utilizó el protocolo *DARC Protein Preparation* de *Scipion.* Con este se eliminaron las moléculas de aguas presentes en la estructura, se eliminó el ligando unido en cada una de las cadenas y se seleccionó la cadena A del archivo pdb. Además, se añadieron a la estructura los hidrógenos y otros átomos no trazados en las cadenas laterales.

III. Mapeado del bolsillo de unión en la proteína y generación de la rejilla electroestática

El bolsillo superficial de NCS-1, que interactua con Ric8a, ha sido mapeado y definido utilizando el protocolo *Generate rays* de *Scipion*, para lo que se seleccionaron los residuos D37 y S93 (Figura 15). Estos residuos fueron seleccionados por la proximidad al residuo R94, que se ha demostrado que participa en la interacción con la proteína Ric8a. Además, se construyó la *grid* electroestática sobre la proteína, usando el protocolo *Grid generation with ADT*. Las dimensiones de la *grid* son de 213 x 213 x 213 puntos, ya que el radio de circunferencia que cubre la proteína des de 40 Å. Se seleccionó un espaciado entre cada punto de la rejilla de 0,375 Å. Así mismo, se asignaron cargas parciales a los átomos de la proteína usando el método según el método de Marsilli-Gasteiger implementado en ADT.

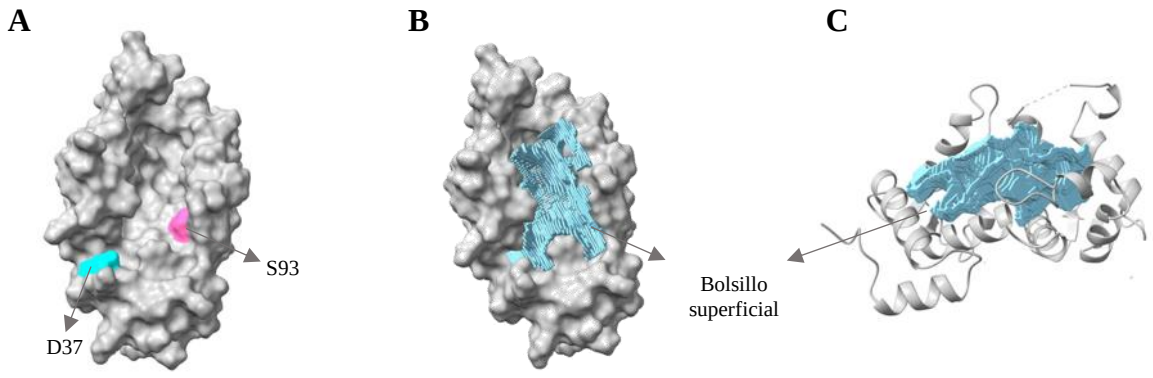


***Figura* 15. A)** Cadena A de la estructura cristalina del NCS-1 humano (PDB: 1G8I), donde mostramos los residuos D37 y S93 utilizados para determinar el bolsillo superficial sobre el que se acoplarán las distintas moléculas de la librería. **B)** En azul, sobre la superficie del NCS-1 (gris) y sobre la estructura secundaria (**C**), se representa el mapeado del bolsillo superficial, al que se unirá Ric8a.

IV. Cribado virtual usando Rosetta DARC

La librería de compuestos y la proteína se sometieron al protocolo de docking en Scipion, *DARC*. El *docking* usando la puntuación basada en la complementariedad de forma y carga por los mejores resultados obtenidos en el caso de uso anterior.

Para ello, en el algoritmo de búsqueda conformacional (PSO), se tuvieron en cuenta 200 partículas y 200 iteraciones por el uso de la búsqueda de confórmeros "*on the fly*". Tras el *docking* las moléculas se clasificaron según la función de puntuación (Tabla S3). El proceso completo de docking necesitó aproximadamente de 1 hora, utilizando la misma GPU que en el caso anterior.

4.2.3 Resultados y discusión

I. Flujo de trabajo en *Scipion*

El flujo de trabajo realizado para este caso de uso de puede visualizar en la Figura 16, donde se llevó a cabo el cribado virtual utilizando cada uno de los pasos mencionados en Materiales y Métodos. En total, se utilizaron 9 protocolos, más el correspondiente a la exportación de los resultados de puntuación, para realizar, de una forma organizada, cómoda, sencilla y trazable, el cribado virtual de una librería de compuestos sobre la superficie de la proteína NCS-1.

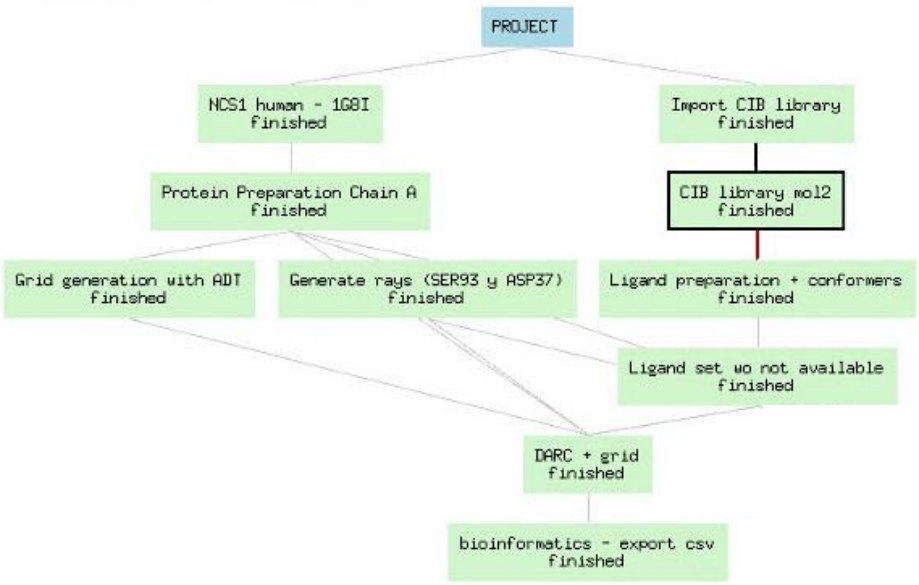


***Figura* 16.** Proyecto de Scipion en el que se ha llevado a cabo el cribado virtual de un total de 1982 compuestos (se descartaron 80 de un total de 2062 compuestos debido a errores en el reconocimiento de átomos por parte de Rosetta en la construcción del archivo de parámetros) con Rosetta DARC.

II. Cribado virtual. Resultados predictivos de Rosetta DARC

El cribado virtual se realizó en el entorno de los residuos S93 y D37 de la proteína humana NCS-1 (PDB: 1G81). En dicho entorno, se encuentran los residuos S93 y R94, que son relevantes en la interacción entre NCS-1 y Ric8a. En este caso, se definió un bolsillo de mayor tamaño que el que se definió en [59], donde solo se realizó el cribado virtual en el entorno de R94. Seleccionamos un bolsillo mayor para buscar compuestos que bloqueen de manera completa y eficaz la superficie de interacción entre NCS-1 y Ric8a, ya que en este experimento se han tenido en cuento casi 1000 compuestos más que los que se tuvieron en cuenta en [59].

De esta forma, con Rosetta DARC se obtuvieron las puntuaciones referidas a la complementariedad de forma y carga entre cada compuesto y NCS-1 en el interior del bolsillo definido (Tabla S3). En el histograma de la Figura 17 se representan la densidad de compuestos para cada rango de valores de puntuación. El rango global de puntuación ocupa desde 5,17359 hasta 718,649, existiendo una mayor presencia de compuestos con puntuaciones entre 0 y 30 y entre 130 y 150. Estos resultados pueden deberse a que la mayoría de los compuestos presentes en el primer rango presentan una estructura similar en un primer análisis visual, algunos de ellos derivados de la aminofenotiazina, como el compuesto SC040 (posición entre los compuestos ordenados por puntuación: 209 y puntuación de 27,645), que ya había sido reportado como un posible inhibidor de la interacción entre NCS-1 y Ric8a [59].

Volviendo a la distribución de puntuaciones para cada compuesto, este amplio rango de puntuaciones entre los compuestos se debe principalmente al mayor tamaño del bolsillo, debido a que no hay apenas penalización por el solapamiento y superación de la forma de éste, al contrario de lo que ocurría en el caso anterior, con un bolsillo de tamaño inferior, donde rápidamente compuestos rígidos y muy grandes (si se habla solo de forma) se penalizaban en gran medida.

Sin entrar en un análisis estructural completo de los complejos proteína-ligando con mejor puntuación (aun sabiendo la existencia de falsos positivos), estos resultados pueden considerarse un punto de partida inicial para filtrar compuestos que, en un ensayo de unión experimental, seguro que iban a ser rechazados, a partir de un análisis tanto de la estructura como de las propiedades farmacológicas más profundas, que sería el siguiente paso para este caso de uso.

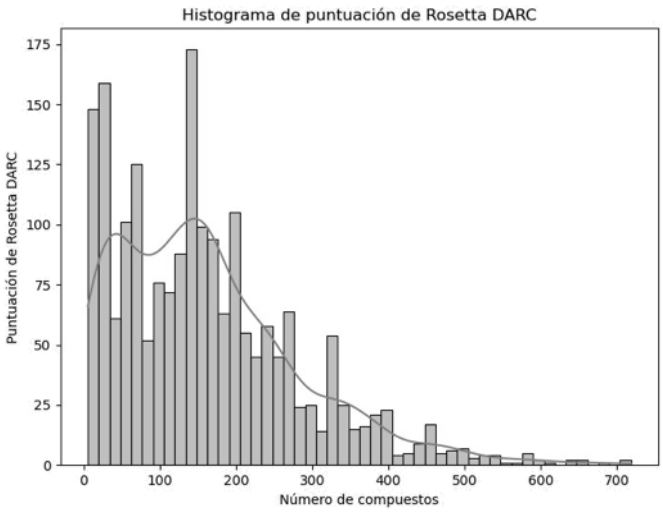


Figura 17. Histograma y curva de densidad de las puntuaciones de Rosetta DARC dada a cada compuesto en base a la complementariedad de carga y forma.

5. Conclusiones

Las conclusiones que se pueden extraer de la integración de Rosetta DARC en *Scipion* y de este trabajo son las siguientes:

1. La integración de distintos programas de *docking* (ej. Rosetta DARC, AUTODOCK, GLIDE), en particular, y quimioinformática (generación de confórmeros y conversión de formatos (ej. OPENBABEL y RDKit, respectivamente)), en general, es posible en la herramienta de protocolos *Scipion*.

2. La integración permite que los flujos de trabajo de cribado virtual de moléculas sean simples, trasparentes y trazables, además de interoperables entre los distintos programas disponibles (de *docking* y de dinámica molecular) para llevar a cabo este tipo de experimentos *in silico*.

3. Para llevar a cabo un proceso de cribado virtual se necesita, como en el caso de Rosetta DARC, una optimización de los parámetros y buena preparación de la estructura de la proteína y moléculas. Si bien es verdad que genera puntos de partida adecuados para análisis estructurales más avanzados, aun presenta un gran número de posibles falsos positivos. Esto se debe al hecho de usar una única función de puntuación basada en el conocimiento (mediante la comparación de topologías principalmente y carga).

4. Rosetta DARC es un programa diseñado para hacer docking proteína-ligando en superficies de PPI, como el caso de la interacción NCS-1/Ric8a. Sin embargo, puede utilizarse, con resultados relativamente aceptables, para cribar compuestos de grandes librerías, como la de ZINC15, para encontrar inhibidores de la actividad ADPr-hidrolasa del Mac1 de la proteína NSP3 del SARS-CoV-2.

6. Perspectivas futuras

Para mejorar los protocolos de cribado virtual y reducir el número de falsos positivos obtenidos con cada programa es necesario, a la vista de los resultados obtenidos en los casos de uso de Rosetta DARC, recurrir a mejoras en la preparación de las proteínas y compuestos, mediante la incorporación de mejores métodos de generación de confórmeros y métodos que establezcan los estados de protonación es tas moléculas a través de métodos como PROPKA.

Así mismo, para mejorar los resultados en el cribado virtual y disminuir la tasa de falsos positivos, sería conveniente la integración de métodos de *docking* con diferentes funciones de puntuación, asegurando la interoperabilidad y la combinación de los distintos métodos, tal y como lo permite Scipion. Estos métodos serían, además de AUTODOCK y GLIDE, UCSF DOCK6, GOLD o HADDOCK. Al utilizar diferentes funciones de puntuación de las uniones predichas, que utilizan una gran diversidad de descriptores de dichas uniones permitirá en casos de uso reales poder seleccionar aquellas uniones predichas computacionalmente que tienen una mayor probabilidad de dar lugar a una interacción real en el laboratorio. Esto sería realizar docking consenso y es el paso siguiente por incorporar en *Scipion*.

Además, el paso siguiente a un cribado virtual sería comprobar la dinámica de las proteínas y estudiar si las uniones predichas por los programas anteriores son lo suficientemente estables. Los programas de dinámica molecular que complementarían a los de *docking* serían GROMACS, AMBER, CHARMM o NAMD.

El objetivo final y este trabajo ha sido el comienzo es construir una plataforma de cribado virtual accesible, fácil de usar, interoperable y trazable por el usuario, sin necesitar de conocimientos avanzados sobre cada uno de los programas usados.

7. Referencias

[1] G. K. Kiriiri, P. M. Njogu, and A. N. Mwangi, "Exploring different approaches to improve the success of drug discovery and development projects: a review," *Futur. J. Pharm. Sci.*, vol. 6, no. 1, pp. 1–12, Dec. 2020, doi: 10.1186/s43094-020-00047-9.

[2] A. Mullard, "2020 FDA drug approvals," *Nature reviews. Drug discovery*, vol. 20, no. 2. NLM (Medline), pp. 85–90, Feb. 01, 2021, doi: 10.1038/d41573-021-00002-0.

[3] O. J. Wouters, M. McKee, and J. Luyten, "Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018," *JAMA - Journal of the American Medical Association*, vol. 323, no. 9. American Medical Association, pp. 844–853, Mar. 03, 2020, doi: 10.1001/jama.2020.1166.

[4] V. Malik, J. K. Dhanjal, A. Kumari, N. Radhakrishnan, K. Singh, and D. Sundar, "Function and structure-based screening of compounds, peptides and proteins to identify drug candidates," *Methods*, vol. 131, pp. 10–21, Dec. 2017, doi: 10.1016/j.ymeth.2017.08.010.

[5] M. Lill, "Virtual screening in drug design," *Methods Mol. Biol.*, vol. 993, pp. 1–12, 2013, doi: 10.1007/978-1-62703-342-8_1.

[6] T. Talele, S. Khedkar, and A. Rigby, "Successful Applications of Computer Aided Drug Discovery: Moving Drugs from Concept to the Clinic," *Curr. Top. Med. Chem.*, vol. 10, no. 1, pp. 127–141, Jan. 2010, doi: 10.2174/156802610790232251.

[7] J. Bajorath, "Pharmacophore," in *Encyclopedia of Cancer*, Springer Berlin Heidelberg, 2008, pp. 2310–2312.

[8] E. H. B. Maia, L. C. Assis, T. A. de Oliveira, A. M. da Silva, and A. G. Taranto, "Structure-Based Virtual Screening: From Classical to Artificial Intelligence," *Frontiers in Chemistry*, vol. 8. Frontiers Media S.A., p. 343, Apr. 28, 2020, doi: 10.3389/fchem.2020.00343.

[9] E. Lionta, G. Spyrou, D. Vassilatis, and Z. Cournia, "Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances," *Curr. Top. Med. Chem.*, vol. 14, no. 16, pp. 1923–1938, Oct. 2014, doi: 10.2174/1568026614666140929124445.

[10] E. Callaway, "Revolutionary cryo-EM is taking over structural biology," *Nature*, vol. 578, no. 7794. NLM (Medline), p. 201, Feb. 01, 2020, doi: 10.1038/d41586-020-00341-9.

[11] M. Batool, B. Ahmad, and S. Choi, "A structure-based drug discovery paradigm," *International Journal of Molecular Sciences*, vol. 20, no. 11. MDPI AG, Jun. 01, 2019, doi: 10.3390/ijms20112783.

[12] M. Hendlich, F. Rippmann, and G. Barnickel, "LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins," *J. Mol. Graph. Model.*, vol. 15, no. 6, pp. 359–363, Dec. 1997, doi: 10.1016/S1093-3263(98)00002-3.

[13] A. T. R. Laurie and R. M. Jackson, "Q-SiteFinder: An energy-based method for the prediction of protein-ligand binding sites," *Bioinformatics*, vol. 21, no. 9, pp. 1908–1916, May 2005, doi: 10.1093/bioinformatics/bti315.

[14] T. Halgren, "New method for fast and accurate binding-site identification and analysis," *Chem. Biol. Drug Des.*, vol. 69, no. 2, pp. 146–148, Feb. 2007, doi: 10.1111/j.1747-0285.2007.00483.x.

[15] T. Sterling and J. J. Irwin, "ZINC 15 - Ligand Discovery for Everyone," *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015, doi: 10.1021/acs.jcim.5b00559.

[16] L. G. Ferreira, R. N. Dos Santos, G. Oliva, and A. D. Andricopulo, “Molecular docking and structure-based drug design strategies,” *Molecules*, vol. 20, no. 7. MDPI AG, pp. 13384–13421, Jul. 01, 2015, doi: 10.3390/molecules200713384.

[17] M. McGann, “FRED pose prediction and virtual screening accuracy,” *J. Chem. Inf. Model.*, vol. 51, no. 3, pp. 578–596, Mar. 2011, doi: 10.1021/ci100436p.

[18] T. J. A. 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. 411–428, 2001, doi: 10.1023/A:1011115820450.

[19] R. A. Friesner *et al.*, “Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy,” *J. Med. Chem.*, vol. 47, no. 7, pp. 1739–1749, Mar. 2004, doi: 10.1021/jm0306430.

[20] G. M. Morris *et al.*, “Software news and updates AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility,” *J. Comput. Chem.*, vol. 30, no. 16, pp. 2785–2791, Dec. 2009, doi: 10.1002/jcc.21256.

[21] O. Korb, T. Stützle, and T. E. Exner, “PLANTS: Application of ant colony optimization to structure-based drug design,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4150 LNCS, pp. 247–258, doi: 10.1007/11839088_22.

[22] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor, “Improved protein-ligand docking using GOLD,” *Proteins Struct. Funct. Genet.*, vol. 52, no. 4, pp. 609–623, Sep. 2003, doi: 10.1002/prot.10465.

[23] W. J. Allen *et al.*, “DOCK 6: Impact of new features and current docking performance,” *J. Comput. Chem.*, vol. 36, no. 15, pp. 1132–1156, Jun. 2015, doi: 10.1002/jcc.23905.

[24] H. Gohlke, M. Hendlich, and G. Klebe, “Knowledge-based scoring function to predict protein-ligand interactions,” *J. Mol. Biol.*, vol. 295, no. 2, pp. 337–356, Jan. 2000, doi: 10.1006/jmbi.1999.3371.

[25] R. S. DeWitte and E. I. Shakhnovich, “SMoG: De novo design method based on simple, fast, and accurate free energy estimates. 1. Methodology and supporting evidence,” *J. Am. Chem. Soc.*, vol. 118, no. 47, pp. 11733–11744, Nov. 1996, doi: 10.1021/ja960751u.

[26] J. D. Durrant and J. A. McCammon, “NNScore 2.0: A neural-network receptor-ligand scoring function,” *J. Chem. Inf. Model.*, vol. 51, no. 11, pp. 2897–2903, Nov. 2011, doi: 10.1021/ci2003889.

[27] M. Wójcikowski, P. J. Ballester, and P. Siedlecki, “Performance of machine-learning scoring functions in structure-based virtual screening,” *Sci. Rep.*, vol. 7, 2017, doi: 10.1038/srep46710.

[28] W. H. Shin and C. Seok, “GalaxyDock: Protein-ligand docking with flexible protein side-chains,” *J. Chem. Inf. Model.*, vol. 52, no. 12, pp. 3225–3232, Dec. 2012, doi: 10.1021/ci300342z.

[29] R. Gowthaman *et al.*, “DARC: Mapping Surface Topography by Ray-Casting for Effective Virtual Screening at Protein Interaction Sites,” *J. Med. Chem.*, vol. 59, no. 9, pp. 4152–4170, May 2016, doi: 10.1021/acs.jmedchem.5b00150.

[30] A. Leaver-Fay *et al.*, “Rosetta3: An object-oriented software suite for the simulation and design of macromolecules,” in *Methods in Enzymology*, vol. 487, no. C, Academic Press Inc., 2011, pp. 545–574.

[31] R. F. Alford *et al.*, “The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design,” *J. Chem. Theory Comput.*, vol. 13, no. 6, pp. 3031–3048, Jun. 2017, doi: 10.1021/acs.jctc.7b00125.

[32] X. Han *et al.*, “Structural insight into catalytic mechanism of PET hydrolase,” *Nat. Commun.*, vol. 8, no. 1, 2017, doi: 10.1038/s41467-017-02255-z.

[33] R. Gowthaman, S. Lyskov, and J. Karanicolas, “DARC 2.0: Improved docking and virtual screening at protein interaction sites,” *PLoS One*, vol. 10, no. 7, p. 131612, Jul. 2015, doi: 10.1371/journal.pone.0131612.

[34] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe, “A fast flexible docking method using an incremental construction algorithm,” *J. Mol. Biol.*, vol. 261, no. 3, pp. 470–489, Aug. 1996, doi: 10.1006/jmbi.1996.0477.

[35] M. A. C. Neves, M. Totrov, and R. Abagyan, “Docking and scoring with ICM: The benchmarking results and strategies for improvement,” *J. Comput. Aided. Mol. Des.*, vol. 26, no. 6, pp. 675–686, Jun. 2012, doi: 10.1007/s10822-012-9547-0.

[36] S. Ruiz-Carmona *et al.*, “rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids,” *PLoS Comput. Biol.*, vol. 10, no. 4, p. 1003571, 2014, doi: 10.1371/journal.pcbi.1003571.

[37] A. Grosdidier, V. Zoete, and O. Michielin, “SwissDock, a protein-small molecule docking web service based on EADock DSS,” *Nucleic Acids Res.*, vol. 39, no. SUPPL. 2, p. W270, Jul. 2011, doi: 10.1093/nar/gkr366.

[38] W. H. Shin, C. W. Christoffer, and D. Kihara, “In silico structure-based approaches to discover protein-protein interaction-targeting drugs,” *Methods*, vol. 131. Academic Press Inc., pp. 22–32, Dec. 01, 2017, doi: 10.1016/j.ymeth.2017.08.006.

[39] H. Lu *et al.*, “Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials,” *Signal Transduction and Targeted Therapy*, vol. 5, no. 1. Springer Nature, pp. 1–23, Dec. 01, 2020, doi: 10.1038/s41392-020-00315-3.

[40] G. Gulfidan, B. Turanli, H. Beklen, R. Sinha, and K. Y. Arga, “Pan-cancer mapping of differential protein-protein interactions,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020, doi: 10.1038/s41598-020-60127-x.

[41] S. M. Vogel *et al.*, “Lithocholic acid is an endogenous inhibitor of MDM4 and MDM2,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 42, pp. 16906–16910, Oct. 2012, doi: 10.1073/pnas.1215060109.

[42] C. Zhuang, S. Narayanapillai, W. Zhang, Y. Y. Sham, and C. Xing, “Rapid identification of Keap1-Nrf2 small-molecule inhibitors through structure-based virtual screening and hit-based substructure search,” *J. Med. Chem.*, vol. 57, no. 3, pp. 1121–1126, Feb. 2014, doi: 10.1021/jm4017174.

[43] L. Borriello *et al.*, “Structure-based discovery of a small non-peptidic Neuropilins antagonist exerting in vitro and in vivo anti-tumor activity on breast cancer model,” *Cancer Lett.*, vol. 349, no. 2, pp. 120–127, Jul. 2014, doi: 10.1016/j.canlet.2014.04.004.

[44] S. Janson, D. Merkle, and M. Middendorf, “Molecular docking with multi-objective Particle Swarm Optimization,” *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 666–675, Jan. 2008, doi: 10.1016/j.asoc.2007.05.005.

[45] K. R. Khar, L. Goldschmidt, and J. Karanicolas, “Fast Docking on Graphics Processing Units via Ray-Casting,” *PLoS One*, vol. 8, no. 8, p. 70661, Aug. 2013, doi:

10.1371/journal.pone.0070661.

[46] J. M. de la Rosa-Trevín *et al.*, "Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy," *J. Struct. Biol.*, vol. 195, no. 1, pp. 93–99, Jul. 2016, doi: 10.1016/j.jsb.2016.04.010.

[47] T. Nakane *et al.*, "Single-particle cryo-EM at atomic resolution," *Nature*, vol. 587, no. 7832, pp. 152–156, Nov. 2020, doi: 10.1038/s41586-020-2829-0.

[48] M. Martínez *et al.*, "Integration of Cryo-EM Model Building Software in Scipion," *J. Chem. Inf. Model.*, vol. 60, no. 5, pp. 2533–2540, May 2020, doi: 10.1021/acs.jcim.9b01032.

[49] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open Babel: An Open chemical toolbox," *J. Cheminform.*, vol. 3, no. 10, pp. 1–14, Oct. 2011, doi: 10.1186/1758-2946-3-33.

[50] S. Geidl *et al.*, "High-quality and universal empirical atomic charges for chemoinformatics applications," *J. Cheminform.*, vol. 7, no. 1, p. 59, Dec. 2015, doi: 10.1186/s13321-015-0107-1.

[51] M. H. M. Olsson, C. R. SØndergaard, M. Rostkowski, and J. H. Jensen, "PROPKA3: Consistent treatment of internal and surface residues in empirical p K a predictions," *J. Chem. Theory Comput.*, vol. 7, no. 2, pp. 525–537, Feb. 2011, doi: 10.1021/ct100578z.

[52] N. M. O'Boyle, T. Vandermeersch, C. J. Flynn, A. R. Maguire, and G. R. Hutchison, "Confab - Systematic generation of diverse low-energy conformers," *J. Cheminform.*, vol. 3, no. 1, pp. 1–9, Mar. 2011, doi: 10.1186/1758-2946-3-8.

[53] "DARC Demo." https://www.rosettacommons.org/demos/latest/public/darc/README (accessed May 31, 2021).

[54] M. Schuller *et al.*, "SARS-CoV-2 identified through crystallographic screening and computational docking," *Sci. Adv.*, vol. 7, no. 16, pp. 8711–8725, Apr. 2021, doi: 10.1126/sciadv.abf8711.

[55] "WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data." https://covid19.who.int/ (accessed Jun. 16, 2021).

[56] C. A. Brosey *et al.*, "Targeting SARS-CoV-2 Nsp3 macrodomain structure with insights from human poly(ADP-ribose) glycohydrolase (PARG) structures with inhibitors," *Prog. Biophys. Mol. Biol.*, vol. 163, pp. 171–186, Aug. 2021, doi: 10.1016/j.pbiomolbio.2021.02.002.

[57] S. Kouhpayeh *et al.*, "The Molecular Basis of COVID-19 Pathogenesis, Conventional and Nanomedicine Therapy," *Int. J. Mol. Sci.*, vol. 22, no. 11, p. 5438, May 2021, doi: 10.3390/ijms22115438.

[58] B. N. Cronstein and T. M. Aune, "Methotrexate and its mechanisms of action in inflammatory arthritis," *Nature Reviews Rheumatology*, vol. 16, no. 3. Nature Research, pp. 145–154, Mar. 01, 2020, doi: 10.1038/s41584-020-0373-9.

[59] A. Mansilla *et al.*, "Interference of the complex between NCS-1 & Ric8a with phenothiazines regulates synaptic function & is an approach for fragile X syndrome," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 114, no. 6, pp. E999–E1008, Feb. 2017, doi: 10.1073/pnas.1611089114.

8. Material suplementario

El material suplementario que se muestra a continuación puede descargarse en la siguiente dirección web, debido a que la Tablas S1, S2 y S3 son tablas de más de 2000 filas cada una.

8.1. Tabla S1: Resultados del *docking* frente al Mac1 de la NSP3 del SARS-CoV-2 teniendo en cuenta la complementariedad de carga y de forma.

8.2. Tabla S2: Resultados del *docking* frente al Mac1 de la NSP3 del SARS-CoV-2 teniendo en cuenta exclusivamente la complementariedad de forma.

8.3. Tabla S3: Resultados del *docking* frente a la proteína NCS-1 teniendo en cuenta la complementariedad de carga y de forma.