

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**USO DE TÉCNICAS COMPUTACIONALES DE ACOPLAMIENTO Y
DE DINÁMICA MOLECULAR PARA PROPONER UN
TRATAMIENTO QUE APLIQUE LA REUTILIZACIÓN DE
MEDICAMENTOS A LA COCCIDIOSIS EN LOS POLLOS DE
ENGORDE**

Tesis para obtener el título profesional de Licenciado en Química

AUTOR:

Santiago Galvez Shoobridge

ASESOR:

Fanny Lys Casado Peña


Lima, Noviembre, 2024

Informe de Similitud

Yo, Fanny Lys Casado Peña, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada: „Uso de técnicas computacionales de Acoplamiento y Dinámica Molecular para proponer un tratamiento que aplique la reutilización de medicamentos a la Coccidiosis en los pollos de engorde“, del autor Santiago Galvez Shooobridge, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 5%. Así lo consigna el reporte de similitud emitido por el software Turnitin el 23/11/2024.
- He revisado con detalle dicho reporte y la Tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 23 de noviembre del 2024.

Apellidos y nombres del asesor: Casado Peña, Fanny	
DNI: 40444557	 Firma:
ORCID: 0000-0002-8791-626X	

Resumen

La coccidiosis es una enfermedad que perjudica la salud del pollo y la producción de su carne, aquella es causada por parásitos del género *Eimeria*. Para combatirla hay un interés de encontrar nuevos medicamentos. Una opción es la reutilización de medicamentos, esto es, usar un fármaco desarrollado para otro propósito.

El objetivo de esta tesis es proponer drogas con posibles efectos terapéuticos. Para lograrlo se usó una técnica de cribado virtual: los medicamentos de una base de datos interactúan mediante técnicas computacionales con una proteína para inhibirla. En otras palabras, las estructuras en 3D de los medicamentos y de la proteína se descargaron de internet para ser usadas en el programa de acoplamiento molecular LeDock. Luego, sólo los medicamentos más prometedores y la proteína se procesaron en el programa de dinámica molecular GROMACS. Finalmente, se analizaron los datos obtenidos.

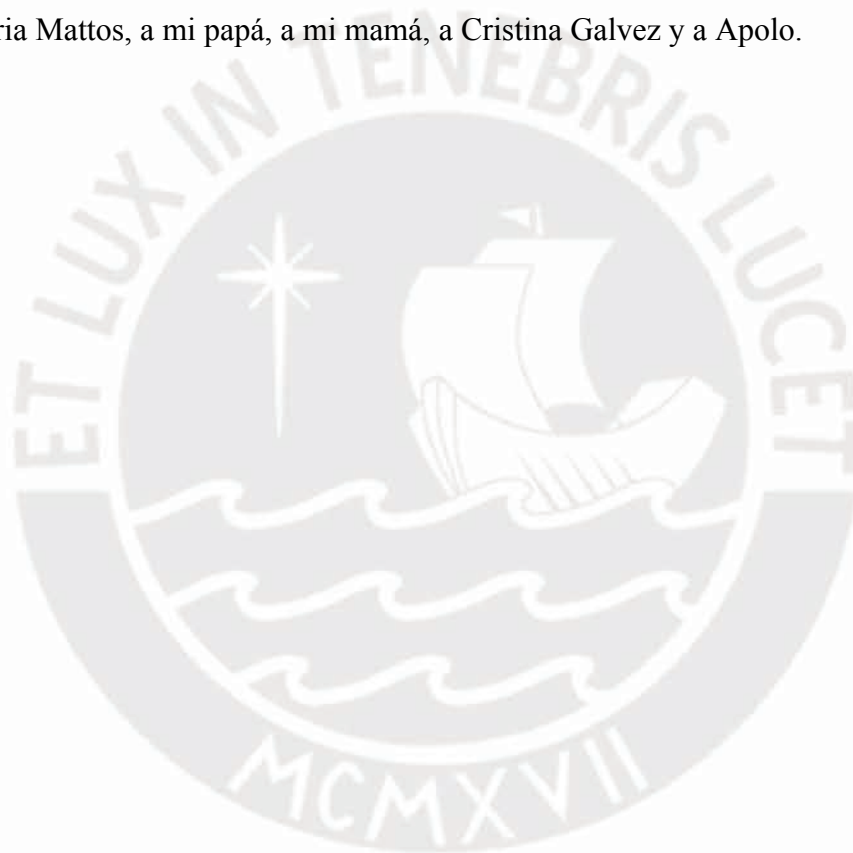
Los resultados indican a dos medicamentos como posibles buenos candidatos para tratar la coccidiosis al inhibir la proteína Dihidroorotato deshidrogenasa de la *E. tenella*. La mejor opción tiene el "PubChem CID" de 89612265, mientras que la otra es el bisantreno, pero todavía falta analizar la viabilidad de ambos para un estudio *in vivo*. Además, debido al bajo poder computacional utilizado y al no usar un mejor programa de acoplamiento molecular, no se lograron encontrar otros candidatos. Sin embargo, se demostró que con bajos recursos computacionales también se pueden obtener resultados y se logró un buen estudio exploratorio que se puede ampliar para otras enfermedades o modificar para otros usos.

Agradecimientos

A mi asesora Fanny Casado, por la supervisión y consejos brindados a lo largo de la elaboración de la tesis.

A los profesores Yulán Hernández, Patricia Gonzales y Luis Ortega que leyeron y corrigieron el manuscrito.

A todos los que me dieron soporte durante la elaboración de la tesis, en particular a Violeta Arana, a Gloria Mattos, a mi papá, a mi mamá, a Cristina Galvez y a Apolo.



Dedicatoria

A Vis, por su apoyo desde los más de seis años que la conozco.



Índice general

Introducción	1
1. La coccidiosis y la reutilización de medicamentos	2
1.1. La carne de pollo: producción y beneficios	2
1.2. La coccidiosis	5
1.3. Desarrollo de nuevos tratamientos y la reutilización de medicamentos	15
1.4. Cribado virtual	21
2. Metodología	27
2.1. Descarga de proteínas y medicamentos	27
2.2. Acoplamiento molecular	29
2.3. Dinámica molecular	29
3. Resultados	31
3.1. Acoplamiento molecular	31
3.2. MMPBSA	34
3.3. RMSD	35
3.4. RMSF	37
4. Discusión y recomendaciones	38
Referencias	42
Anexos	48
A.1. Programas y computadoras	48
A.1.1. Computadoras	48
A.1.2. Programas	49
A.1.3. Instalación	50
A.2. Información complementaria del procedimiento	53
A.2.1. Búsqueda en ChEMBL	53
A.2.2. Acoplamiento molecular	54
A.2.3. Trayectorias de DM	60
A.3. Archivos “.sh”	61
A.3.1. a.sh	61
A.3.2. b.sh	67
A.3.3. c.sh	70
A.3.3. 2b.sh	70
A.4. Resultados completos	73

Índice de figuras

Figura 1:	Producción mundial de carne pecuaria	3
Figura 2:	Precios de la carne pecuaria	3
Figura 3:	Ciclo de vida de <i>Eimeria</i> spp.	7
Figura 4:	Monensina unida a un catión	13
Figura 5:	Etapas del desarrollo de medicamentos	17
Figura 6:	Error de AM	32
Figura 7:	Antes y después de DM	33
Figura 8:	RMSD: ferulenol y co-factores	35
Figura 9:	RMSD representativos	36
Figura 10:	RMSF	37
Figura 11:	Candidatos encontrados	40



Índice de tablas

Tabla 1: Comparación de FCRs	4
Tabla 2: <i>Eimeria</i> spp. del pollo	8
Tabla 3: Archivos de AM: valores de afinidad	31
Tabla 4: Valores de afinidad: MMPBSA	34
Tabla 5: CPUs de las laptops	48
Tabla 6: Desempeño de las laptops	49



Abreviaturas

AM	: acoplamiento molecular
CPU	: unidad central de procesamiento, <i>por sus siglas en inglés</i>
DM	: dinámica molecular
FCR	: índice de transformación del alimento, <i>por sus siglas en inglés</i>
GPU	: unidad de procesamiento gráfico, <i>por sus siglas en inglés</i>
HTS	: cribado de alto rendimiento, <i>por sus siglas en inglés</i>
MMPBSA	: Mecánica Molecular Poisson-Boltzmann Área de Superficie, <i>por sus siglas en inglés</i>
RMSD	: raíz de la desviación cuadrática media, <i>por sus siglas en inglés</i>
RMSF	: raíz de la fluctuación cuadrática media, <i>por sus siglas en inglés</i>
spp.	: todas o múltiples especies del género en mención
USD	: dólares estadounidenses, <i>por sus siglas en inglés</i>



Introducción

La producción de carne de pollo tiene una enorme importancia económica y alimentaria a nivel mundial (FAOSTAT, 2023a; FAOSTAT, 2023b; Orkusz, 2021). Una de las principales enfermedades que afectan a la salud del pollo y a su producción de carne es la coccidiosis (Gustin & McCarter, 2021; McDougald et al., 2020). El objetivo de esta tesis es proponer drogas con posibles efectos terapéuticos para combatirla.

En el presente trabajo, se empleó una técnica de cribado virtual que consiste en evaluar de manera computacional la interacción de los medicamentos de una base de datos con una proteína, específicamente la Dihydroorotato deshidrogenasa de la *E. tenella*, la cual es necesaria para la supervivencia del parásito causante de la enfermedad (Sato et al., 2020).

Esta tesis consta de cuatro capítulos. El primero comienza con la importancia económica de la carne de pollo a nivel mundial. Luego, explica la coccidiosis en el contexto de que es la principal enfermedad que afecta a la producción de carne de pollo. A continuación, se menciona como se puede combatir esta enfermedad con fármacos desarrollados para otros propósitos. Por último, se mencionan métodos computacionales para encontrar estos medicamentos. El segundo capítulo consiste del procedimiento realizado para proponer los fármacos mencionados. Este incluye el uso de los métodos computacionales de acoplamiento y dinámica molecular mediante los programas LeDock y GROMACS, respectivamente. Mientras que el tercer capítulo presenta los resultados más relevantes del procedimiento seguido, los cuales incluyen información del comportamiento y afinidad de los medicamentos con la proteína que se quiere inhibir. Finalmente, en el cuarto capítulo, se discute acerca de los candidatos obtenidos, de las mejoras y de las limitaciones de los programas utilizados y de sus posibles aplicaciones.

Capítulo 1

1. La coccidiosis y la reutilización de medicamentos

La coccidiosis es una enfermedad que afecta a la producción de carne de pollo, la cual tiene patógenos resistentes a todos los tratamientos actuales (Chapman, 2014; Gustin & McCarter, 2021). Debido a esto, hay mucho interés en encontrar nuevos medicamentos (Noack et al., 2019). Un método rápido y barato es utilizar un fármaco ya desarrollado para otra aplicación, esto es, la reutilización de medicamentos (Panic et al., 2014).

Por lo tanto, los subcapítulos de esta parte van a abordar, en el siguiente orden, información general acerca de la carne de pollo, las características de la coccidiosis, la importancia de la reutilización de medicamentos y, por último, los dos métodos computacionales de acoplamiento y de dinámica molecular, estos pueden ser usados en hallar medicamentos candidato para un nuevo tratamiento de la coccidiosis.

1.1. La carne de pollo: producción y beneficios

Respecto a la producción de carne pecuaria, en la Figura 1 se muestra como la carne de pollo tiene un aumento constante y más pronunciado que los otros tipos de carne a nivel mundial. Es más, supera a la de ganado vacuno en el año 1999 y a la de chanco, en el año 2019. Cabe añadir que sólo se muestran tres fuentes de carne pecuaria, pero juntas representan entre 87% y 89% del total. De igual manera, el pollo representa entre el 85% y 90% de las aves de corral. Además, el consumo de carne de pollo ha aumentando debido a varios factores relacionados. Algunos de estos son su crianza industrial, su índice de transformación del alimento (FCR) y el precio de su carne (Figura 2).

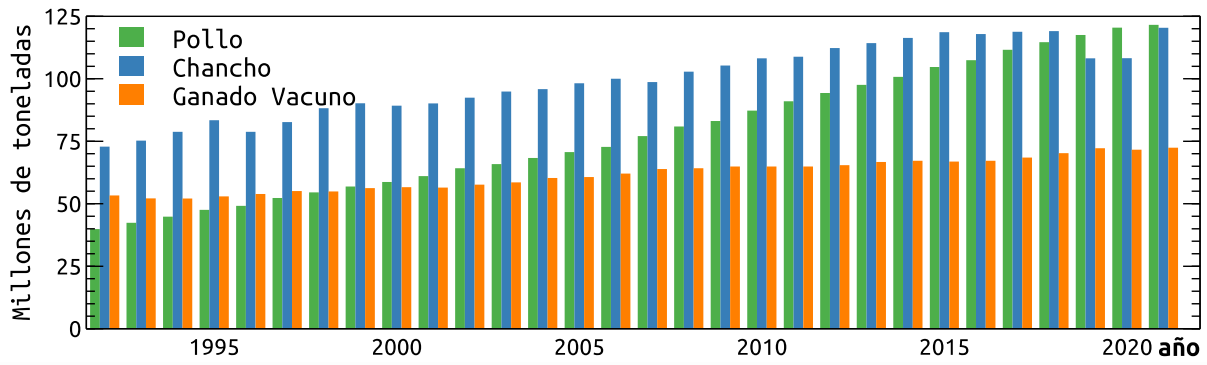


Figura 1 Producción mundial de carne pecuaria. Se muestran las producciones anuales, del año 1992 al año 2021, de tres tipos de carne. Elaborada con datos de FAOSTAT, 2023b.

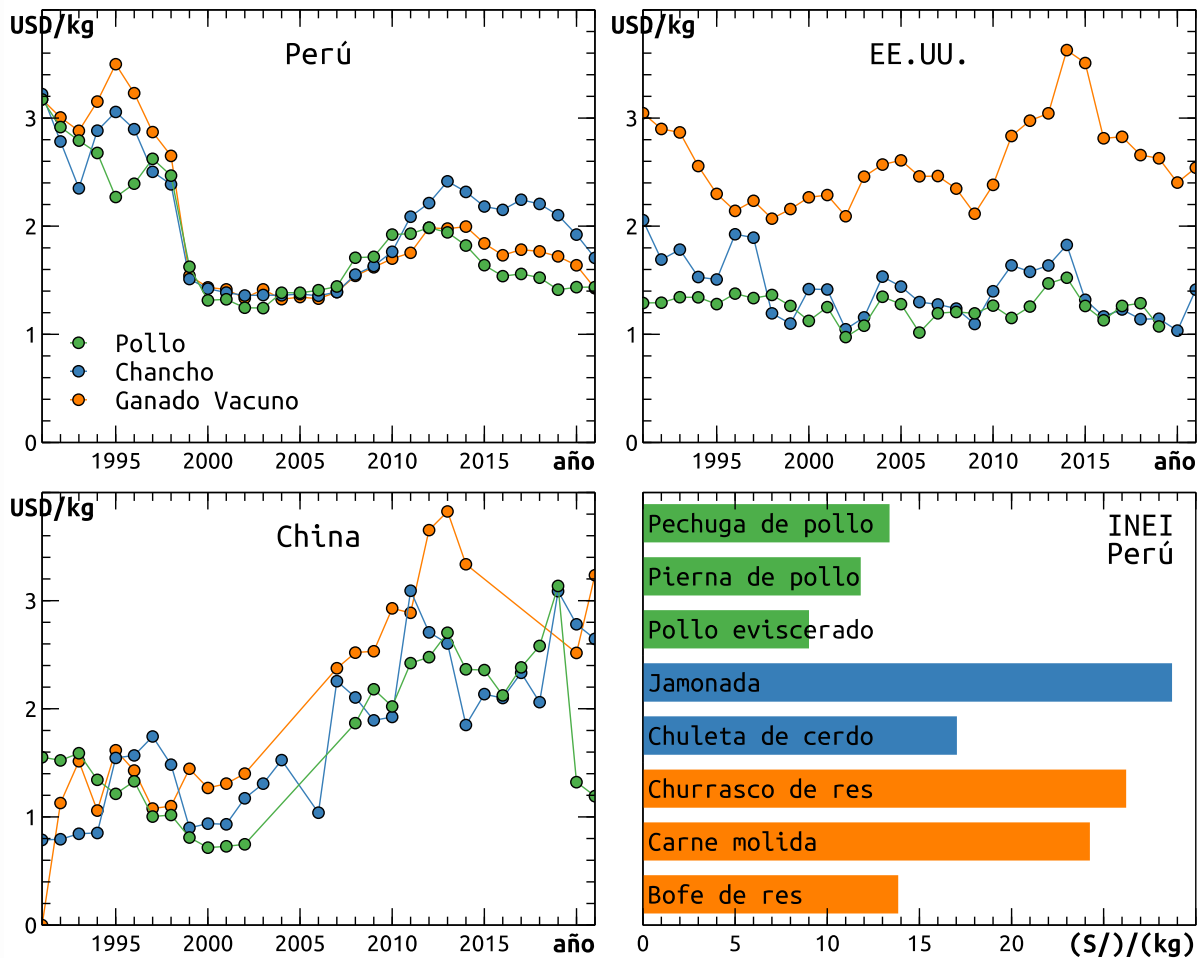


Figura 2 Precios de la carne pecuaria. En los tres primeros recuadros se muestra el precio, en centros de acopio, del animal completo por kilogramo en dólares estadounidenses (USD) del 2020. El último recuadro muestra precios para los consumidores en el Perú respecto al año 2021. Elaborada con datos de FAOSTAT, 2023a; INEI, 2022; Webster, 2023.

Por un lado, la crianza industrial se ve favorecida por el uso de criaderos climatizados para controlar la temperatura, la humedad, la ventilación, la luz, entre otros factores, y por el uso de variantes extremadamente especializadas que tienen un alto rendimiento para producir exclusivamente carne o huevos, estas son conocidas como pollos de engorde y gallinas ponedoras, respectivamente (Bennett et al., 2018; Mueller et al., 2020). Además, son matadas a una edad temprana (es decir, los pollos de engorde viven de 5 a 7 semanas y las gallinas ponedoras, 1 año), son criadas en altas densidades (por ejemplo, 18 pollos/m²) y tienen una diversidad genética reducida (Bennett et al., 2018; Bergeron et al., 2020; Cobanoglu et al., 2014; Li et al., 2021; Mueller et al., 2020). Este manejo industrial puede perjudicar necesidades futuras, ya que vuelve a los pollos más vulnerables a enfermedades (McDougald et al., 2020; Muir et al., 2008).

Por otro lado, el FCR es el peso del pienso dividido por el aumento de peso (Fry et al., 2018). Sin embargo, este valor es impreciso porque no considera cuál es la parte comestible, qué nutrientes tiene esta ni cómo es la retención de proteínas y de calorías. Como se aprecia en la Tabla 1, el pollo tiene un mejor FCR que el chancho y que el ganado vacuno.

Tabla 1 Comparación de FCRs. Sólo se están utilizando valores de Fry et al., 2018.

Animales	FCR	Pienso/comestible (en peso)
Ganado vacuno	8,0	14,0
Chancho	3,9	5,3
Pollo	1,9	2,6

Respecto al valor nutricional, la carne en general es una fuente de proteína completa, esto es, contiene todos los aminoácidos esenciales, y otros más, en cantidades equilibradas, una fuente de grasa y una fuente de hierro de fácil digestión (Orkusz, 2021). Sin embargo, la carne de pollo, comparada con otras carnes, es una mejor fuente de proteína y de grasas: en particular, la pechuga de pollo es buena para complementar raciones diarias con deficiencia de proteína y su ratio de ácidos grasos poliinsaturados a saturados está entre los mejores

(Ibrahim et al., 2018; Orkusz, 2021; Strobel et al., 2012). Una desventaja es su menor contenido de hierro (de 0,4 a 1,3) respecto al de las carnes rojas (3,1 para la carne de res y 3,5 para la de caballo, valores expresados en mg/100 g) (Orkusz, 2021).

Acerca de la salud del pollo, hay que tener en cuenta la enfermedad que se describe a continuación.

1.2. La coccidiosis

El filo Apicomplexa incluye a los géneros *Eimeria*, *Toxoplasma* (por ejemplo, *T. gondii* que causa la toxoplasmosis), *Plasmodium* (responsables de la malaria), *Isospora*, *Hemoproteus*, *Leucocytozoon*, *Sarcocystis*, *Wenyonella*, *Tyzzeria* y *Cryptosporidium* (McDougald et al., 2020; Noack et al., 2019). Este filo está compuesto de parásitos obligados intracelulares que abarcan a los responsables de la coccidiosis, la cual es una enfermedad infecciosa que depende de los sistemas de producción. Por ejemplo, se agrava al criar a pollos en un espacio reducido, debido a que los parásitos tienen ciclos de vida cortos y un huésped específico, esto es, las *Eimeria* spp. que afectan al pollo no infectan a otros animales (Chapman et al., 2010; McDougald et al., 2020).

La coccidiosis afecta a animales salvajes y domésticos, y se caracteriza por diarrea, morbilidad, pérdida de peso, deshidratación y, a veces, mortalidad (McDougald et al., 2020; Noack et al., 2019). Asimismo, los parásitos que la generan alteran la mucosa intestinal a medida que se reproducen (McDougald et al., 2020). Esto a su vez provoca una absorción reducida de nutrientes, la pérdida de proteínas del plasma por filtración, la alteración en la pigmentación de la piel y una mayor susceptibilidad a otras enfermedades que pueden agravar a la coccidiosis (Chapman, 2014; McDougald et al., 2020). Por ejemplo, en aves de corral, la coccidiosis favorece la colonización de bacterias dañinas como la *Clostridium*

perfringens y de parásitos como el *Histomonas meleagridis* (McDougald et al., 2020).

Es importante estudiar a la enfermedad para tenerla bajo control, para mejorar el bienestar animal y para encontrar tratamientos más accesibles y beneficiosos. En ese sentido, respecto al pollo, a continuación, se abordarán los costos de la enfermedad, luego, las características de los parásitos que la generan y, finalmente, los tratamientos disponibles.

1.2.1. Costos

Respecto al impacto industrial, en cada uno de los expedientes de las reuniones anuales de la *United States Animal Health Association* del 2016 al 2019, la coccidiosis, en particular la causada por *E. maxima*, fue clasificada como la enfermedad más importante de los pollos de engorde, mientras que la enteritis necrótica fue la segunda más importante. Estas dos enfermedades normalmente operan en conjunto (Burlson, 2019; Gustin & McCarter, 2021; Johnson, 2017; Mountainspring & Burlson, 2018). Aunque es complicado calcular los costos que genera la coccidiosis en pollos de engorde, se presentan dos cálculos a continuación.

Uno es respecto al Reino Unido en el año 1995. El costo de la enfermedad fue de 4,6% con respecto al precio del pollo. Teniendo en cuenta que la ganancia neta fue igual al 7,0% del precio del pollo, el costo de la enfermedad, respecto a la ganancia neta, fue de 66% (Williams, 1999).

El otro cálculo es respecto al año 2016, pero a nivel mundial. En este caso, el costo de la coccidiosis fue de alrededor del 4,9% del precio de venta, con lo cual, el costo total fue de 13,9 mil millones de USD, aunque, si se incluye a la industria de huevos de gallina, el costo asciende a 14,1 mil millones de USD (Blake et al., 2020; Exchange Rates UK, 2021; FAOSTAT, 2023a, 2023b).

Para reducir estos costos significativos y prevenir que aumenten, es necesario conocer la enfermedad. Por lo tanto, se van a describir las características de los patógenos que la generan, incluida su propagación.

1.2.2. Patógeno (*Eimeria* spp.)

En el caso de los pollos, los parásitos causantes de la coccidiosis son del género *Eimeria* (Chapman et al., 2010; McDougald et al., 2020). En la Figura 3, se muestra su ciclo de vida.

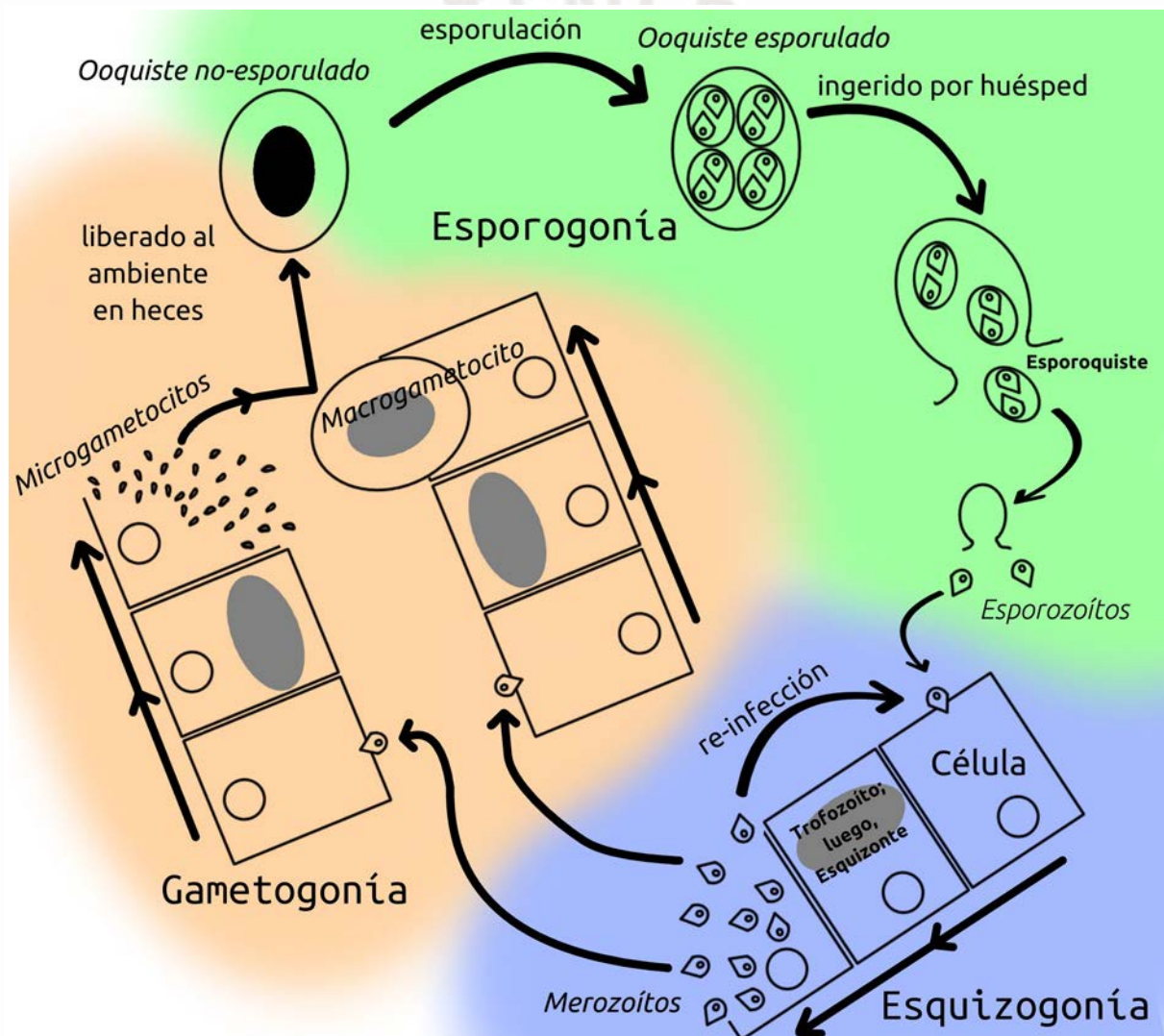


Figura 3 Ciclo de vida de *Eimeria* spp. Se muestran las 3 etapas del ciclo: la esporogonía que dura hasta la liberación de esporozoítos, la esquizogonía que tiene un número determinado de re-infecciones (el número depende de la especie), y la gametogonía que corresponde a la única etapa de reproducción sexual y dura desde los merozoítos de la última generación hasta la producción del ooquiste no-esporulado. Elaborada con información de McDougald et al., 2020; Wikimedia, 2007.

Con respecto a las especies de este género que afectan a los pollos, en la Tabla 2 se muestran las siete especies de *Eimeria* spp. reconocidas mundialmente (Chapman, 2014; Chapman et al., 2010; McDougald et al., 2020). Cabe añadir que normalmente las infecciones por *E. acervulina*, *E. praecox*, *E. maxima*, *E. mitis* y *E. tenella* se observan de las 3 a las 6 semanas de edad, las de *E. necatrix* de las 8 a las 18 y las de *E. brunetti* se ven temprano y tarde. Por lo tanto, las gallinas reproductoras y las ponedoras están más expuestas porque viven de 20 semanas a más (McDougald et al., 2020).

Tabla 2 *Eimeria* spp. del pollo. Se muestran las diferentes especies de *Eimeria* y tres características de cada una. Respecto a la gravedad de las lesiones, A representa hemorragia o muerte, B, inflamación intestinal con hemorragias puntuales, C, pérdida de fluidos y mala absorción de nutrientes y D, poca patogenicidad. Información obtenida de Chapman, 2014; McDougald et al., 2020.

Especie	Ubicación gastrointestinal donde se desarrollan (parte superior (s), parte inferior (i) y lugares secundarios)	Gravedad de las lesiones	Dimensiones de ooquistes ($\mu\text{m}\times\mu\text{m}$)
<i>E. acervulina</i>	duodeno, yeyuno, íleon	C	18,3×14,6
<i>E. praecox</i>	duodeno, yeyuno	D	21,3×17,1
<i>E. necatrix</i>	yeyuno, íleon (s), ciegos	A	20,4×17,2
<i>E. maxima</i>	duodeno, yeyuno, íleon (s), íleon (i)	B	30,5×20,7
<i>E. mitis</i>	íleon, ciegos	C	15,6×14,2
<i>E. tenella</i>	íleon (i), ciegos, colon	A	22,0×19,0
<i>E. brunetti</i>	íleon (i), colon, resto de intestinos	B	24,6×18,8

Respecto al diagnóstico, este debe basarse en necropsias representativas, en las lesiones y en encontrar diferentes etapas del parásito. No es suficiente hallar algunos ooquistes en el intestino, esto solo indica la presencia de la infección, mas no la gravedad ni si está controlada (McDougald et al., 2020). Además, parte del diagnóstico es dar información acerca de cuáles *Eimeria* spp. están causando daño y cuánto, para elegir un tratamiento adecuado (McDougald et al., 2020; Noack et al., 2019).

Por un lado, para diferenciar las especies suele ser adecuado utilizar sus características tradicionales: el lugar que abarca la infección en el tracto gastrointestinal, la gravedad de las lesiones y el tamaño de diferentes etapas, como el de los ooquistes o el de los esquizontes. Por ejemplo, la *E. necatrix* tiene esquizontes de 17,3 μm de largo, mientras que los de la *E. maxima* miden 9,4 μm . Además, la primera es una pobre productora de ooquistes (los

cuales se desarrollan en los ciegos), por lo cual tiene una baja capacidad reproductiva (McDougald et al., 2020).

Sin embargo, se debe tener en cuenta que la infección por dos o más especies de coccidios es común y, para una confirmación adicional, se requieren estudios de inmunidad cruzada y bioquímicos como la reacción en cadena de la polimerasa. Este, además de determinar las especies, da información acerca de la contribución por especie (Chapman, 2014; McDougald et al., 2020).

Por otro lado, la ingesta de ooquistes esporulados es el único método natural de transmisión y la infección depende del número ingerido (Chapman, 2014; McDougald et al., 2020; Noack et al., 2019). Estos suelen ser más numerosos en los excrementos del criadero cuando la edad del pollo es de 3 a 5 semanas (McDougald et al., 2020). Además, la esporulación depende de la temperatura donde la óptima está entre 25 a 30°C, la humedad, la cantidad de oxígeno dado que el proceso usa enzimas aeróbicas y la especie (Chapman, 2014; McDougald et al., 2020; Noack et al., 2019). Esta suele demorar de 24 a 72 h, aunque la esporulación de la *E. praecox* puede llegar a ocurrir en 12 h. También hay que tener en cuenta el tiempo desde la ingesta del ooquiste hasta la primera expulsión: el menor es de 3 d 11 h, mientras que el mayor es de 5 d 18 h, y corresponden a la *E. praecox* y a la *E. necatrix*, respectivamente (McDougald et al., 2020).

Cabe añadir que la manera más común de propagar la enfermedad es el transporte mecánico de los ooquistes debido al tránsito del personal, aunque también puede ser por insectos (por ejemplo, moscas), equipos contaminados, aves salvajes y polvo (McDougald et al., 2020).

Por lo tanto, para combatir a la enfermedad, es indispensable evitar fugas de agua, tener ventilación adecuada, limitar el acceso humano y emplear medidas de higiene (Noack et al., 2019). En un clima cálido y seco, la amenaza de coccidiosis es menor (McDougald et al.,

2020).

Es importante tener en cuenta que no se ha logrado la eliminación ni la prevención de coccidiosis mediante cuarentena, desinfección o saneamiento, ya que el patógeno está prácticamente en todos los criaderos y sus ooquistes son extremadamente resistentes (Chapman, 2014; McDougald et al., 2020). Además, un entorno estéril podría permitir brotes futuros desastrosos, debido a que el sistema inmunológico del huésped no va estar preparado para la enfermedad (McDougald et al., 2020; Parker et al., 2018).

Respecto a los ooquistes, su pared celular tiene dos capas: la interna está basada en una glicoproteína y la externa, en 1-hexacosanol (Chapman, 2014). Estos resisten a los desinfectantes comunes y a condiciones ambientales extremas: para matarlos se necesita congelarlos, secarlos, o exponerlos a 55°C o, durante dos a tres días, a 37°C. A pesar de que pueden sobrevivir durante muchas semanas en el suelo, después de que las aves se retiran de la granja, su supervivencia en el criadero se limita a unos pocos días debido al calor y al amoníaco liberados por la cama (McDougald et al., 2020).

Teniendo en cuenta que prácticamente no se puede eliminar a la enfermedad y a las medidas mencionadas para evitar su propagación, a continuación, se van a mencionar formas de combatir a la coccidiosis.

1.2.3. Tratamientos

Prácticamente siempre es necesario utilizar tratamientos preventivos, lo cual se debe a dos razones: primero, es demasiado tarde para evitar consecuencias negativas cuando los signos clínicos son evidentes y, segundo, existe el riesgo de un brote desastroso de la enfermedad si no hay profilaxis (McDougald et al., 2020; Noack et al., 2019). En contraste, los tratamientos curativos sólo se usan como último recurso (McDougald et al., 2020).

Un tema que hay que tener en cuenta sobre los tratamientos, que incluyen al uso de medicamentos y a la vacunación, es la resistencia. La resistencia cruzada ocurre cuando la resistencia generada para un tratamiento permite resistir otros tratamientos, ya que la modificación producida en el organismo protege contra ambos debido a alguna similitud compartida, por ejemplo, mecanismos de acción similares entre medicamentos. Mientras que la resistencia múltiple se define como varias resistencias generadas individualmente a cada tratamiento al que fue expuesto el organismo (Chapman et al., 2010). Estos dos tipos de resistencia no son excluyentes entre sí.

Debido a la capacidad que tienen las *Eimeria* spp. de generar resistencia, se necesita restaurar la sensibilidad de estos parásitos a los tratamientos que la combaten (Chapman, 2014; Chapman et al., 2010; Noack et al., 2019). Existen dos maneras de restaurarla, las cuales se pueden usar en conjunto. Una es alternar los tratamientos durante la vida de los pollos o cada vez que un grupo de pollos sale del criadero: estos no deben generar resistencia cruzada. Por ejemplo, se pueden alternar medicamentos con otros de diferente mecanismo de acción o con vacunas (Chapman, 2014; Chapman et al., 2010). La otra es inocular a los pollos con parásitos no-resistentes: estos se reproducen con los resistentes y las nuevas generaciones de parásitos recuperan sensibilidad (Chapman et al., 2010). Es más, se pueden usar parásitos atenuados mediante selección artificial para que sus esquizontes sean más pequeños y sólo tengan una generación, y que esta produzca una menor cantidad de merozoítos (Chapman, 2014).

Por otro lado, el uso de vacunas no sólo se ve favorecido por la resistencia generada de las *Eimeria* spp. a los medicamentos, si no también por el rechazo a estos para poder etiquetar el producto final como “completamente natural”. Sin embargo, la mayoría de vacunas usan ooquistes vivos, las cuales tienen la desventaja de que producen infecciones leves que afectan

negativamente el aumento de peso, la conversión alimenticia y la pigmentación de la piel. Además, la vacunación no logra eliminar el uso de medicamentos, el cual sigue siendo fundamental para el control de la coccidiosis. Por ejemplo, aunque hay productores que usan etiquetados restrictivos como “producción orgánica certificada” o más permisivos como “criado sin antibióticos”, los criadores se pueden ver forzados, para evitar la muerte de sus pollos, a utilizar medicamentos que los obliguen a prescindir de esos etiquetados (McDougald et al., 2020).

1.2.4. Medicamentos

En el caso de los anticoccidiales, estos se dividen en dos categorías: ionóforos y compuestos sintéticos (Noack et al., 2019).

Los ionóforos usados en el tratamiento de la coccidiosis son productos de fermentación por *Streptomyces* spp. o por *Actinomadura* spp. y corresponden a la monensina, a la salinomicina, a la narasina, a la lasalocida, a la maduramicina y a la semduramicina (Noack et al., 2019). Respecto a su mecanismo de acción, cuando estos atrapan cationes, generan una entidad de coordinación con grupos apolares en la superficie exterior (Figura 4) (Chapman et al., 2010; Noack et al., 2019). Esto a su vez genera un proceso de difusión pasiva en membranas lipídicas, por ejemplo, las membranas celulares, mitocondriales y de algunos orgánulos. Lo cual descompensa al patógeno, ya que altera sus gradientes iónicos (Chapman et al., 2010).

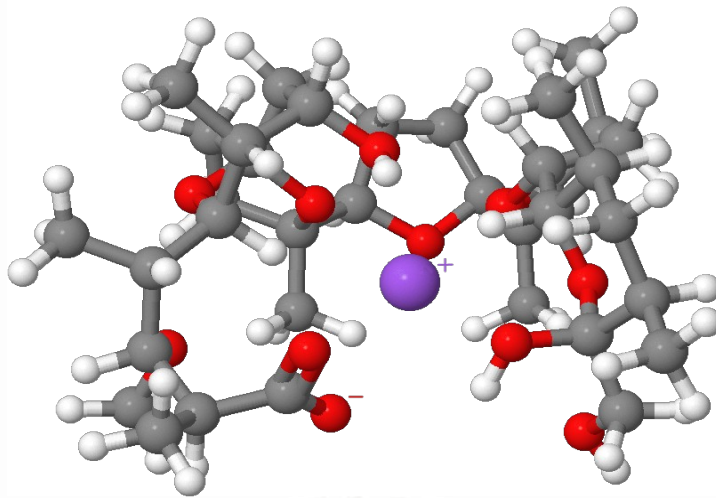


Figura 4 Monensina unida a un catión. Ejemplo de un ionóforo (monensina) con carga neta 1- que rodea a un catión con carga neta 1+ (de color morado), lo cual forma una entidad de coordinación neutra con una superficie exterior apolar. La energía no está optimizada. Basada en Chapman et al., 2010.

Este mecanismo contribuye a que la resistencia sea difícil de inducir y de desarrollo lento, ya que el parásito se adapta al cambiar su pared celular, lo cual demora en ocurrir (Chapman, 2014). Sin embargo, eventualmente se genera resistencia. Por ejemplo, en el caso de la monensina, su dosis es de 20 a 40 veces mayor para los parásitos resistentes respecto a los sensibles (Chapman et al., 2010).

También, el mecanismo de acción de los ionóforos permite que tengan un amplio espectro contra diferentes *Eimeria* spp., es decir, detienen o matan al esporozoíto, al merozoíto o al trofozoíto joven (Chapman, 2014; Noack et al., 2019). Sin embargo, genera efectos negativos como, por ejemplo, daño mitocondrial, el cual influye en degeneración muscular incluida la cardíaca y en neuropatía. Esto genera un margen de seguridad estrecho: hay peligro con errores en la dosis y con exposición a otras especies. Otro ejemplo, es su incompatibilidad con algunos antioxidantes, con algunas sulfonamidas y con varios antibióticos usados para tratar infecciones bacterianas, por ejemplo, tiamulina, cloranfenicol, eritromicina, y oleandomicina (Noack et al., 2019).

Por otro lado, los compuestos sintéticos que más han durado en la industria son el amprolio, el zoaleno y la nicarbazina, ordenados de mayor a menor probabilidad de que las

Eimeria spp. generen resistencia. Otros compuestos, como el diclazurilo, la halofuginona, el decoquinato, el clopidol y la robenidina, son más propensos a producir resistencia. Debido a esto, dos de los anteriores, el clopidol y la halofuginona, dejaron de ser comerciales a partir del 2017 (Collett et al., 2020).

El amprolio es un inhibidor competitivo de la absorción de tiamina, afecta a esquizontes de segunda generación de *E. Tenella* y tiene una gran ventana de seguridad, por lo cual está aprobado para prevención y tratamiento de gallinas ponedoras (Chapman, 2014; McDougald et al., 2020; Noack et al., 2019).

El zoaleno y la nicarbazina destruyen a los esquizontes de la primera o de la segunda generación (McDougald et al., 2020). Sin embargo, falta dilucidar sus mecanismos de acción, aunque se cree que el segundo inhibe la fosforilación oxidativa (Chapman, 2014; Noack et al., 2019). Además, ambos pueden generar ataxia y la nicarbazina aumenta el riesgo de golpe de calor, es altamente tóxica para gallinas ponedoras y tiene una pequeña ventana de seguridad (Fulton, 2020; Noack et al., 2019).

Respecto al espectro de actividad, el amprolio se mezcla con etopabato (más activo contra *E. maxima* y *E. Brunetti* pero carece de actividad contra *E. Tenella*, por lo cual se complementan bien) o con sulfaquinoxalina (una sulfonamida), ambos son antagonistas del ácido fólico, esto es, son análogos estructurales de este o de un precursor (Noack et al., 2019). En contraste, la nicarbazina tiene un amplio espectro pero se puede mezclar con la narasina o la monensina, ya que tiene sinergia con estas (Chapman, 2014; Chapman et al., 2010). Asimismo, la monensina tiene sinergia con la lasalocida, la nicarbazina, el clopidol y la roxarsona (Chapman et al., 2010). La última pertenece a las drogas arsenicales, discontinuadas en muchos países desde el 2015 (Noack et al., 2019). Cabe mencionar que el clopidol y el decoquinato (una quinolona) son de amplio espectro e inhiben la respiración

mitocondrial del parásito, pero las *Eimeria* spp. desarrollan rápidamente resistencia a estos. Por ejemplo, los citocromos del parásito se modifican para que las quinolonas no se unan tan fácilmente a estos (Noack et al., 2019).

La mayoría de compuestos sintéticos son antiguos o, comparados con los ionóforos y las vacunas, caros. Además, los compuestos sintéticos son más propensos a generar resistencia respecto a los ionóforos y a las vacunas (McDougald et al., 2020). Es más, actualmente hay parásitos de *Eimeria* spp. resistentes a todos los medicamentos usados para combatir la coccidiosis, por lo que se necesita quimioterapia preventiva, novedosa y rentable (Chapman, 2014; Noack et al., 2019). En ese sentido, los siguientes capítulos de esta sección van a abordar el desarrollo de nuevos medicamentos.

1.3. Desarrollo de nuevos tratamientos y la reutilización de medicamentos

Para los humanos y otros animales, se necesitan nuevos medicamentos y terapias contra parásitos eucariotas unicelulares, ya que los tratamientos actuales suelen tener varias limitaciones como poca o ninguna eficacia contra algunas etapas del parásito, un margen de seguridad estrecho, e ineficacia en huéspedes jóvenes o inmunodeprimidos (Abenoja et al., 2021). Esto también es cierto para el caso específico de las *Eimeria* spp. que afectan a los pollos.

En general, hay que tener presente los siguientes puntos durante el desarrollo de un nuevo tratamiento y durante su uso.

- ▶ El espectro de actividad, es decir el rango en el que mata o inhibe al patógeno sin ser muy tóxico para el huésped (McDougald et al., 2020).
- ▶ La eficacia, la cual se puede calcular, por ejemplo, al medir cambios en la mortalidad, en el aumento de peso, en las lesiones intestinales o en la producción de ooquistes (Chapman et al.,

2010).

► El mecanismo de acción, el cual sirve para optimizar el uso e incluye saber si el metabolito es el compuesto activo (Hughes et al., 2011; Noack et al., 2019).

► Su efecto sobre el patógeno, por ejemplo, las etapas específicas del parásito afectadas por un medicamento (Chapman, 2014; McDougald et al., 2020).

► El desarrollo de resistencia que se cuantifica mediante el tiempo que demora en aparecer (Noack et al., 2019).

► El método de aplicación, el cual puede ser mediante alimentación. Por ejemplo, medicamentos y vacunas, los cuales pueden ser añadidos en el agua para un efecto más rápido, aunque requiere una mayor precaución respecto a la preparación y a la administración (Chapman, 2014; McDougald et al., 2020). También, puede ser mediante rociado, el cual es el caso de algunas vacunaciones por ejemplo (Chapman, 2014). Otras posibilidades incluyen usar transportadores moleculares que mejoren las propiedades del tratamiento, por ejemplo, mejorar la accesibilidad al lugar de actividad, o una mayor o una menor permanencia en el organismo (Abenoja et al., 2021; Yin et al., 2021). En cualquier caso hay que tener en cuenta que tan práctico es aplicar el tratamiento, ya que esto puede determinar su preferencia respecto a otro o un uso adecuado del mismo (Woods et al., 2011).

► La toxicidad hacia otras especies, particularmente la humana, y hacia el medio ambiente, incluye la generación de resistencia en cualquier patógeno (Noack et al., 2019; Reis et al., 2020a, 2020b).

Cabe añadir que, para combatir la coccidiosis, se pueden atacar las características únicas del parásito con respecto al huésped. Por ejemplo, estos tienen un orgánulo en particular, el apicoplasto, que sintetiza ácidos grasos durante la esquizogonía y la gametogonía. Además, los ooquistes usan amilopectina y manitol como reservas de energía (Chapman, 2014). Dicho

esto, hay que tener en cuenta las etapas del desarrollo de nuevos medicamentos.

1.3.1. Todas las etapas del desarrollo de medicamentos

Hay varias estrategias para encontrar nuevos medicamentos. Una es desarrollar medicamentos desde la identificación de una característica que se pueda atacar para combatir a la enfermedad (Abenoja et al., 2021; Hughes et al., 2011; Singh & Seed, 2021). Las primeras etapas, para este proceso, son la identificación de compuestos que podrían interactuar con dada característica y la validación de los mismos, esto es, detectar si son activos (Hughes et al., 2011; Singh & Seed, 2021).

Luego, viene la etapa clínica. Para llegar a esta, usualmente se requiere de mucho tiempo y de la contribución de varias disciplinas y profesionales. Además, muy pocos de los compuestos llegan a esta etapa y los que salen también son escasos, esto es, alrededor del 10% en ambos casos. Esto se debe a que son rechazados principalmente por ineficacia o por toxicidad (Hughes et al., 2011). Finalmente, luego de ser aprobado un medicamento, viene su comercialización y seguimiento (Hughes et al., 2011; Singh & Seed, 2021; A. Wong, 2019).

El proceso mencionado puede variar, pero es importante tener un concepto básico de su estructura, este se muestra en la Figura 5.

Etapas	Preclínica		Clínica				
	Investigación básica	Estudios	Fase I	Fase II	Fase III	Aprobación	Fase VI
	síntesis objetivos cribados	eficacia toxicidad pruebas en animales	seguridad en 20 a 100 personas sanas	seguridad y eficacia en 100 a 500 pacientes	seguridad y eficacia respecto a otras drogas en más de 500 pacientes	por la entidad correspondiente	comercialización y seguimiento

Figura 5 Etapas del desarrollo de medicamentos. Es respecto al uso humano, los pasos se pueden superponer y pueden haber procesos que abarquen más de uno, por ejemplo, el desarrollo de la producción comercial puede comenzar desde los estudios preclínicos. Basada en Ciani & Jommi, 2014; FDA, 2022; SEIKAGAKU, 2021.

Cabe añadir que durante todas las distintas etapas se pueden usar herramientas computacionales (Hughes et al., 2011; Woods et al., 2011). Por ejemplo, se pueden usar en

asistir la decisión de detener o continuar el desarrollo del nuevo medicamento, al calcular los costos y la probabilidad de que tenga éxito en la etapa clínica (Hughes et al., 2011). Esta es la etapa más cara y es más perjudicial cancelarla, en parte porque hacerlo puede, según Hughes et al., 2011, ahuyentar a la inversión.

Otro ejemplo es el uso de algoritmos para analizar grandes bases de datos y generar información que pueda ser entendida al ser compacta, abstracta o específica, por ejemplo, un informe breve, una aproximación descriptiva o un modelo predictivo, respectivamente (Fayyad et al., 1996). Esto puede permitir agrupar compuestos similares, descubrir relaciones entre estructura y actividad o efectos secundarios, y predecir el comportamiento de modificaciones o el costo de la síntesis. Lo anterior puede ser utilizado para optimizar la etapa de identificación al hacer una selección más focalizada. También, el uso de conocimientos teóricos puede ayudar en este proceso (Hughes et al., 2011).

Estas herramientas van a depender de las bases de datos que se tengan. Por ejemplo, las bibliotecas virtuales disponibles y sus características, que pueden incluir estructuras de rayos-X. También, van a depender de los algoritmos que se utilicen, por ejemplo, los que permiten predecir modificaciones útiles alrededor de una estructura molecular base o calcular interacciones con algún ligando de interés conocido (Hughes et al., 2011).

Cabe añadir que en las etapas de identificación y validación, conviene combinar múltiples pruebas que incluyen experimentos *in vitro*, *in vivo*, e *in silico*. El objetivo es ir encontrando cada vez mejores medicamentos mediante cribados o la modificación de estructuras moleculares, entre otros. Por ejemplo, está el cribado de alto rendimiento (HTS), el cual usa placas con una gran cantidad de pocillos, esto es, más de 380, de los cuales se usa uno por candidato (Hughes et al., 2011).

En cada uno de estos pocillos puede haber de manera constante una proteína, un tejido vivo,

células o parásitos. También, lo anterior puede ser lo variable y lo constante puede ser un medicamento propuesto, ya sea una molécula sintética o un extracto natural. Respecto a los extractos naturales, estos son una mezcla de sustancias cuyo principio activo puede ser una o varias moléculas que actúan de manera sinérgica. Además, son difíciles de patentar, lo cual desincentiva su investigación (Jayawardene et al., 2021).

Los HTS suelen ser costosos pero, para contrarrestar esa desventaja, se puede hacer un filtrado previo con otras técnicas, por ejemplo, minería de datos o cribado virtual. Aunque usar estructuras conocidas puede dificultar conseguir una patente (Hughes et al., 2011).

Otro ejemplo de cribado es el cribado virtual, el cual consiste en examinar una enorme base de datos de compuestos (por ejemplo, medicamentos) en busca de la afinidad de cada uno de estos con una diana, por ejemplo, una proteína o parte de su estructura (Salmaso & Moro, 2018).

Una ventaja de los experimentos *in silico* es que permiten reducir el uso de las pruebas *in vitro* e *in vivo* (Hughes et al., 2011; Jayawardene et al., 2021). Esta última tiene las desventajas de requerir una mayor cantidad de compuesto, un mayor tiempo y el trabajo de criar animales, las cuales pueden subir los costos además de dilemas éticos (Jayawardene et al., 2021; Russow, 1999; Singh & Seed, 2021). Sin embargo, las técnicas *in vivo*, en los animales objetivo, son más confiables que las *in vivo* en modelos animales y que las *in vitro* e *in silico* (Jayawardene et al., 2021; Singh & Seed, 2021).

En este sentido, respecto al desarrollo de medicamentos para el uso humano y para el uso veterinario, es importante tener en cuenta sus diferencias y sus similitudes. Por ejemplo, la medicina veterinaria tiene la ventaja de poder realizar estudios en las especies de destino durante la etapa preclínica. Sin embargo, tiene las desventajas de ser menos rentable y de tener algunos requisitos adicionales como garantizar que no se pase el nivel máximo de

residuos en productos destinados para el consumo humano, esto está relacionado con la dosis permitida y el tiempo previo necesario para detener el consumo del medicamento (Woods et al., 2011).

En general, para el desarrollo de nuevos medicamentos, se necesita infraestructura y conocimiento. Por ello, es importante desarrollar una buena comunicación, que incluya la transferencia de información (por ejemplo, a través de bases de datos y de compartir conocimiento) entre el sector industrial que tiene recursos para desarrollar HTS y el sector académico (Hughes et al., 2011). Sobre todo porque producir un medicamento exitoso desde las primeras etapas del proceso de desarrollo (es decir, desde sospechar de una manera de combatir una enfermedad con una o más sustancias y detectar su actividad biológica) suele ser costoso y suele durar varios años (Abenoja et al., 2021; Grinnan et al., 2019; Hughes et al., 2011; Singh & Seed, 2021). Sin embargo, tiene la ventaja de una mayor probabilidad en obtener una patente (excepto si la sustancia a validar proviene de la naturaleza) en comparación con otras estrategias como la reutilización de medicamentos que suele ser más barata y suele requerir menos tiempo (Grinnan et al., 2019; Jayawardene et al., 2021).

1.3.2. La reutilización de medicamentos

Esta tiene un significado amplio que abarca continuar con compuestos que han fallado en alguna etapa del proceso de desarrollo y ampliar la utilidad de medicamentos, existentes o ya aprobados, mediante el desarrollo de un tratamiento para pacientes graves, el uso en una enfermedad diferente, un cambio de dosis, nuevas combinaciones o nuevos métodos de aplicación (Abenoja et al., 2021; Grinnan et al., 2019; Panic et al., 2014).

Además, tiene la ventaja de acelerar el proceso de desarrollo debido a menores costos, a menores riesgos y a un menor tiempo de comercialización (Panic et al., 2014). Ya que puede

haber información previa, como los perfiles de seguridad, y a que en algunos casos pueden pasar directamente a alguna fase de la etapa clínica (Huang et al., 2020; Panic et al., 2014). También puede haber infraestructura para su producción y el medicamento puede ser más barato (Grinnan et al., 2019). Esto a su vez permite encontrar tratamientos para enfermedades descuidadas que en el sector privado no son rentables investigar (Sohraby et al., 2019).

Para la reutilización de medicamentos es útil comprender la enfermedad y la farmacología de la medicación reutilizada. Sin embargo, los ejemplos más exitosos de esta han sido descubiertos sin un enfoque sistemático (como es el caso del sildenafil y del minoxidilo que fueron desarrollados para problemas cardíacos e hipertensión y terminaron usándose en disfunción eréctil y alopecia, respectivamente). Esto incluye los descubrimientos accidentales debido a la observación de efectos secundarios, lo cual se puede sistematizar, por ejemplo, con minería de datos en registros de salud electrónicos. Para este caso y en general, es mejor cuando las bases de datos tienen información completa y varios parámetros, ya que permite hacer más comparaciones (Alaimo & Pulvirenti, 2019; Grinnan et al., 2019).

Aparte de técnicas como la minería de datos, se puede usar el cribado virtual. A continuación, se va a profundizar acerca de este método y luego de su uso en una estrategia particular para la reutilización de medicamentos.

1.4. Cribado virtual

Los métodos de cribado virtual aprovechan las estructuras experimentales de bases de datos de proteínas o de medicamentos. Estos se basan en que el conocimiento de la estructura de la diana puede ayudar a interpretar, predecir y optimizar las interacciones ligando-diana, ya que están mediadas por su complementariedad. Además, debido a que el complejo formado puede tener más de una conformación (particularmente en la unión), hay un interés continuo en

mejorar la flexibilidad de los enlaces en las simulaciones. Esto implica un aumento en el número de grados de libertad del sistema y, por lo tanto, en el esfuerzo computacional (Salmaso & Moro, 2018).

A continuación, se explica las técnicas utilizadas en la parte experimental. La primera se utilizó para un cribado virtual y la segunda para analizar los resultados de este. Ambas, al igual que otras técnicas computacionales, utilizan a la mecánica molecular con el fin de reducir el coste computacional necesario respecto a los cálculos que requiere la mecánica cuántica (Salmaso & Moro, 2018). La mecánica molecular realiza cálculos aproximados mediante campos de fuerza (esto es, conjuntos de funciones), los cuales toman en cuenta las fuerzas inter- e intra- moleculares ajustadas mediante valores mecánicos cuánticos o experimentales (Salmaso & Moro, 2018; van de Waterbeemd et al., 1997). En pocas palabras, los átomos se consideran como esferas cargadas conectadas por resortes (Salmaso & Moro, 2018).

1.4.1. Acoplamiento molecular (AM)

Este tiene como objetivo predecir las uniones con mejor afinidad entre cada ligando, por ejemplo, medicamentos de una base de datos, y un objetivo, por ejemplo, una proteína. Esto se logra mediante la generación de una serie de posibles poses del ligando dentro de la región de unión a la proteína, es decir, el sitio activo. Por ello, la disponibilidad de las estructuras tridimensionales son importantes. Respecto a la proteína, esta se puede obtener de manera experimental, mediante cristalografía de rayos X o resonancia magnética nuclear por ejemplo, o de manera computacional como el modelado por homología. Además, el AM (*Molecular docking*, en inglés) se compone principalmente de dos etapas: la orientación de las moléculas y la puntuación de las poses obtenidas (Salmaso & Moro, 2018).

Respecto a la orientación, hay tres tipos de acoplamiento, cada uno es más preciso que el anterior pero requiere un mayor cálculo computacional (Salmaso & Moro, 2018).

► **Rígido:** tanto el “ligando” como la proteína se consideran entidades rígidas, es decir, sólo se considera a la traslación y a la rotación, con tres grados de libertad para cada uno. Se utiliza principalmente para el acoplamiento proteína-proteína.

► **Semi-flexible:** sólo una de las moléculas, el ligando, es flexible, mientras que la proteína es rígida, esto es, el ligando tiene los seis grados de libertad anteriores y, también, los conformacionales.

► **Flexible:** se basa en el concepto de que una proteína no es una entidad rígida pasiva durante la unión, por lo que considera tanto al ligando como a la proteína como contrapartes flexibles. Debido a su mayor costo computacional, se requiere un mejor balance entre precisión y velocidad, lo cual también es cierto para la etapa de puntuación.

Acerca de la puntuación para la afinidad entre el ligando y la proteína, prácticamente hay tres categorías no excluyentes de funciones (Salmaso & Moro, 2018).

► **Basadas en campos de fuerza:** conjuntos de funciones para modelar el comportamiento molecular del sistema.

► **Empíricas:** Se usa la suma de varios términos de energía experimental, ponderados mediante coeficientes, los cuales han sido optimizados para generar datos de afinidad.

► **Basadas en conocimiento:** la afinidad se calcula mediante una base de datos de estructuras y la premisa de que las uniones ligando-proteína más comunes van a tener interacciones más favorables.

Por ejemplo, el programa LeDock utiliza acoplamiento semi-flexible y campos de fuerza. Los resultados de este programa pueden ser utilizados en la siguiente técnica computacional.

1.4.2. Dinámica Molecular (DM)

La dinámica molecular es un procedimiento de simulación computacional que calcula, usualmente mediante campos de fuerza, el movimiento de átomos (van de Waterbeemd et al., 1997). Para generar un entorno más real, esta usa una caja de simulación que trata de manera flexible a los elementos que contiene, por ejemplo, ligando, proteína y solvente. La estrategia que usa es calcular las fuerzas que actúan sobre los átomos y la aceleración que generan en tiempos muy cortos, por ejemplo, de 2 femtosegundos (Salmaso & Moro, 2018).

Debido a esto, el costo computacional es elevado. Antes se requería el uso de una super-computadora, pero ahora se puede usar una estación de trabajo, es decir, *workstation* en inglés. Esto permite reducir los costos y se debe al uso de las unidades de procesamiento gráfico (GPUs), las cuales pueden realizar miles de cálculos en paralelo. Además, hay un mayor acceso a la DM mediante servicios de computación en la nube (Salmaso & Moro, 2018).

Hay que tener en cuenta que, en una simulación de DM, el sistema está estancado en un mínimo, esperando un evento que supere a una barrera de energía. Una estrategia es usar numerosas simulaciones en paralelo con un mismo punto de partida y, cuando en una se escapa del mínimo energético, esa se vuelve el nuevo punto de partida. Otra estrategia es añadir una fuerza al sistema, mediante el empleo de variables en todo el sistema, como la temperatura por ejemplo, o en partes específicas, por ejemplo, el ligando (Salmaso & Moro, 2018).

Además, la DM puede generar información acerca de la estabilidad, de la actividad y de la afinidad, por ejemplo, mediante la raíz de la desviación cuadrática media (RMSD), la raíz de la fluctuación cuadrática media (RMSF) y la Mecánica Molecular Poisson-Boltzmann Área de Superficie (MMPBSA), respectivamente (Sohraby et al., 2019).

Acerca de la RMSD, está relacionada con la distancia promedio entre átomos: los del esqueleto de la proteína respecto a los del ligando. Se puede construir un gráfico de RMSD *versus* tiempo. Este sirve para analizar la estabilidad del complejo, ya que el complejo es estable cuando los valores de RMSD varían menos de 0,1 nm, es decir, se asemejen a una recta paralela al eje del tiempo, y cuando son inferiores a cierto valor, por ejemplo, 0,5 nm. Además, fluctuaciones grandes indican cambios estructurales y la gráfica permite ubicar el tiempo exacto en el que suceden. Para este análisis conviene que la simulación de DM sea de 100 ns o mayor (Sohraby et al., 2019).

Por otro lado, la RMSF mide la distancia promedio del movimiento de algún átomo durante un periodo de tiempo. Por ejemplo, el de un carbono- α durante el último nano-segundo de la simulación. Por lo tanto, se puede crear un gráfico de estos valores de RMSF *versus* su respectivo carbono- α . Esto va a dar información sobre cuánto se mueve cada parte de la proteína, lo cual está relacionado con su funcionamiento. En otras palabras, si el espectro con un ligando se parece al que tiene un inhibidor y es diferente al de la proteína sola, puede que el ligando tenga una función inhibidora similar (Sohraby et al., 2019).

Respecto a la MMPBSA, este método permite calcular la afinidad, entre el ligando y la proteína, mediante energías (por ejemplo, de van der Waals, electrostáticas, de solvatación polar y del área superficial accesible por el solvente) de diferentes momentos extraídos de la simulación (Aldeghi et al., 2017; Sohraby et al., 2019). En pocas palabras, el resultado del cálculo es igual a la energía libre del complejo menos la del ligando y la de la proteína ($\Delta G_{\text{unión}} = G_{\text{complejo}} - G_{\text{ligando}} - G_{\text{proteína}}$) (Sohraby et al., 2019). Sin embargo, valores de afinidad más precisos y confiables se pueden obtener con otros métodos, pero estos suelen tardar más tiempo y ser más complejos en su ejecución (Aldeghi et al., 2017; Sohraby et al., 2019). Por último, en relación con el AM, el método MMPBSA redefine la puntuación de afinidad a

valores más reales (Sohraby et al., 2019).

Un programa importante para realizar simulaciones de DM es GROMACS, el cual también permite realizar RMSD, RMSF y MMPBSA, aunque para el último se necesita instalar un programa adicional. GROMACS junto con LeDock y con otros programas son descritos en el capítulo siguiente sobre metodología.

Respecto a la parte experimental de esa tesis, cabe añadir que se analizaron 4 241 medicamentos en LeDock, que luego se procesaron los 14 más representativos en GROMACS, y que mediante los datos obtenidos se lograron los objetivos específicos de encontrar dos buenos candidatos, uno con “PubChem CID” de 89612265 y el otro es el bisantreno, de corroborar computacionalmente la inhibición *in vitro* del ferulenol observada por Sato et al., 2020, y de generar un estudio exploratorio en el uso de las técnicas computacionales en cuestión para encontrar medicamentos que traten enfermedades, a pesar de que se requieren mejoras en los programas y en el procedimiento. En resumen, se cumplió el objetivo general de proponer drogas con posibles efectos terapéuticos para tratar a la coccidiosis.

Capítulo 2

2. Metodología

Se emplearon técnicas computacionales y la estrategia de Sohraby et al., 2019 con el objetivo de encontrar buenos candidatos, es decir, medicamentos ya existentes, para inhibir una proteína de *Eimeria* spp. Lo cual puede permitir desarrollar un tratamiento para la coccidiosis que afecta a la producción de carne de pollo.

Cabe mencionar que este capítulo, el siguiente y los anexos usan información complementaria que no se puede guardar adecuadamente en este documento, como videos, comandos y hojas de cálculo. Para referirse al lugar donde esta guardada esta información se va a usar DRIVE, ya que esta en un “*Google Drive*”:

<https://drive.google.com/drive/folders/1GfJQK3uXxswAdbbJ8X0g4R3RnzACxYp-?usp=sharing>. Además, un duplicado está en <https://doi.org/10.5281/zenodo.13773972>.

Por otro lado, el procedimiento realizado consta de (1) descarga de estructuras 3D, (2) Acoplamiento molecular y (3) Dinámica molecular. Además, requiere el uso de los programas descritos en el anexo “A.1.”, estos incluyen a LeDock, a GROMACS y a ChimeraX para AM, DM y visualización de moléculas, respectivamente.

2.1. Descarga de proteínas y medicamentos

2.1.1. Proteínas

Varias estructuras de proteínas se pueden obtener de <https://www.rcsb.org/>. Se buscó “eimeria” en esta página web para encontrar proteínas de este parásito. Se descargó la “6AJE” en formato “.pdb” de RCSB Protein Data Bank, 2019, también se puede descargar en formato “.ent”: estos archivos son idénticos (es decir, se puede editar el nombre para intercambiar las terminaciones). También se descargó “5E34” de RCSB Protein Data Bank,

2015 para comparar en LeDock.

El “6AJE” es la estructura cristalina, obtenida por difracción de rayos X, de un complejo formado por una proteína con sus dos co-factores y con un inhibidor. La proteína es la Dihidroorotato deshidrogenasa de la *E. tenella* y pertenece a su membrana mitocondrial. Los co-factores son flavín mononucleótido (FMN) y orotato (ORO), mientras que el inhibidor es ferulenol. Sin embargo, una parte de la proteína no pudo ser dilucidada por su baja densidad electrónica (Sato et al., 2020).

2.1.2. Medicamentos

Las estructuras de los medicamentos se descargaron de <https://pubchem.ncbi.nlm.nih.gov/> (PubChem) en SDF 3D, esto es, un archivo “.sdf” con varias estructuras 3D dentro. Por ejemplo, en el 2023-04-25, se obtuvo un archivo “.sdf” que corresponde a la categoría “Drug and Medication Information/Clinical Trials/13 809” con el filtro “Molecular Weight: de 2,016 a 1002” (las diferentes categorías están en <https://pubchem.ncbi.nlm.nih.gov/classification/#hid=72>).

Cabe mencionar que en AM se seleccionaron los medicamentos que contengan sólo los átomos C, N, H y O, hasta 5 oxígenos, y que tengan valores mayores a 200 Da, A este grupo se le va a llamar G_PubChem. Se filtraron de la manera descrita porque se obtiene una buena cantidad de medicamentos y porque se espera que estos sean más baratos y que se demoren menos en simular.

También se obtuvieron otros “.sdf”: en el 2022-07-05, se utilizó <https://www.ebi.ac.uk/chembl/> (ChEMBL) para buscar medicamentos y PubChem para descargarlos (ver anexo “A.2.1.” para mayor detalle). A este grupo se le va a llamar G_ChEMBL.

En AM se usaron 2 646 y 2 333 medicamentos de G_ChEMBL y de G_PubChem, respectivamente, aunque entre estos dos grupos hay 739 duplicados. Además, se añadieron dos estructuras de ferulenol, esto es, la de 6AJE y una descargada de PubChem. En otras palabras, se procesaron 4 242 estructuras de medicamentos en total.

2.2. Acoplamiento molecular

Primero, se trabajó con el archivo de la proteína, este es 6aje.pdb. A partir de este se crearon las carpetas MD_6aje y Docking_6aje. La primera contiene a los archivos MD.pdb, cof_fmn.pdb, cof_oro.pdb y l_ferulenol.pdb, mientras que la otra, a los archivos pro.pdb y dock.in.

Luego, se prepararon los medicamentos: se usaron los archivos “.sdf” para crear archivos “.mol2” y guardarlos en Docking_6aje. Después, en esa carpeta, se usó LeDock para obtener las posiciones de los medicamentos con sus puntajes de afinidad y para crear archivos “.pdb” de las posiciones de los medicamentos con mayor afinidad a la proteína. Finalmente, se copiaron los mejores archivos “.pdb” en MD_6aje. Los pasos detallados están en “A.2.2.”.

En resumen, a partir de la proteína y los medicamentos descargados, se obtuvieron con LeDock las posiciones de acoplamiento de los medicamentos con sus puntajes respectivos. Y, después, se usó esa información para usar las mejores posiciones en dinámica molecular.

2.3. Dinámica molecular

Para esta parte se usaron a la carpeta MD_6aje y a 3 archivos “.sh” de texto plano: “a.sh”, “b.sh” y “c.sh”. El contenido de los “.sh” se muestra en el anexo “A.3.”: en sus comentarios (comienzan con “#”) están los pasos a seguir y algunas descripciones acerca de lo que hacen los comandos. Además, estos archivos están en el DRIVE en la carpeta “Comandos”.

Hay que tener en cuenta que para correr b.sh, sólo se necesita haber instalado GROMACS (es decir, “b) GROMACS 2022.3”), mientras que para correr a.sh y c.sh es necesaria la instalación completa para DM. Esto permite correr b.sh, la parte más demandante de GROMACS, en computadoras que se dediquen sólo a eso.

Al correr estos archivos “.sh”, se genera una carpeta c_f a partir de archivos “.pdb” de la proteína a trabajar (es decir, MD.pdb), de los co-factores y de los medicamentos. Para el caso de 6aje, c_f pesa alrededor de 5,85 GiB por medicamento. Dentro de esta carpeta hay otra, de nombre “R”, que contiene a la carpeta “csv”. En esta última van a estar los archivos em_PE.csv, rmsd.csv, rmsf.csv y mmpbsa.csv.

El primero de estos archivos sólo está para corroborar si el paso de minimizar la energía del complejo fue adecuado. Los dos siguientes se usan para crear las gráficas RMSD (ns vs nm) y RMSF (número de C-alpha vs nm), respectivamente. Mientras que el archivo mmpbsa.csv da nuevos valores de la afinidad del medicamento a la proteína: un menor valor en kcal/mol (más negativo) indica una unión más fuerte entre ambos.

Por otro lado, las trayectorias de las simulaciones, que están en la carpeta R, se pueden visualizar si se abre su respectivo archivo [L]_md.gro en ChimeraX y, luego, su trayectoria. Estas son, para el primero y para el último 1 ns, [L]_i1s.xtc y [L]_f1s.xtc, respectivamente, “[L]” representa el nombre del medicamento. El último nano-segundo suele ser más importante, porque puede representar un sistema estable. El anexo “A.2.3.” explica como generar trayectorias, incluidas las que estén en un momento determinado de la simulación.

En resumen, se usaron las mejores posiciones, según y obtenidas por LeDock, de los medicamentos para generar, mediante GROMACS, simulaciones de su comportamiento. Estas, a su vez, generan las gráficas RMSD y RMSF, y nuevos valores de la afinidad del medicamento a la proteína, es decir, lo obtenido mediante MMPBSA.

Capítulo 3

3. Resultados

Como se menciona anteriormente, se analizaron 4 240 medicamentos más dos estructuras de ferulenol en AM. Para diferenciarlas, las que comienzan con A o B van a corresponder a G_ChEMBL y las que comienzan con C, a G_PubChem. Además, luego de la letra viene el código de PubChem del medicamento y después la posición en “.pdb” obtenida por AM usada, por ejemplo, C_89612265_16, A_135564890_01 y B_42613186_02. Además, en DM se analizaron dos estructuras de ferulenol: la primera es la quinta mejor posición generada por AM de la estructura obtenida por 6AJE, la cual se analizó sin y con co-factores, mientras que la otra corresponde a la primera posición de la estructura descargada. Sus códigos son F_54679300_05_SinCof, F_54679300_05_ConCof y Fd_54679300_01, respectivamente.

Primero, se van a mencionar lo obtenido mediante AM y, luego, lo de dinámica DM. La información de este último corresponde a las posiciones finales de DM (al final del capítulo de AM), a los valores de afinidad obtenidos mediante MMPBSA y a los gráficos de RMSD y RMSF.

3.1. Acoplamiento molecular

Los valores de afinidad obtenidos se pueden ver en los seis archivos del DRIVE > Resultados > AM, sus nombres están en la Tabla 3.

Tabla 3 Archivos de AM: valores de afinidad

Archivo	Medicamentos	co-factores	Proteína
6aje_AyB_LeDock.csv 6aje_AyB.csv	G_ChEMBL	sin co-factores	
6aje_CHNO5y200aMLeDock.csv			6AJE
6aje_CHNO5y200aM_LeDock_d.csv 6aje_SinCof.csv	G_PubChem	con co-factores	
H5N1_5e34_CHNO5y200aM.csv			5E34

Los que contienen en su nombre “AyB” y el archivo 6aje_SinCof.csv corresponden a G_ChEMBL y a los ferulenoles. Solamente el último archivo mencionado se corrió sin co-factores, ya que se realizaron pruebas con y sin co-factores en AM: cuando hay co-factores los valores de energía son más negativos, pero se mantiene un orden similar y las posiciones de acoplamiento son prácticamente equivalentes.

Por otro lado, los que contienen en su nombre “CHNO” corresponden a G_PubChem. Además, cinco de los seis archivos se corrieron con la proteína 6AJE: el restante es H5N1_5e34_CHNO5y200aM.csv y se corrió con 5E34. Este se comparó con 6aje_CHNO5y200aMLeDock.csv para detectar sesgos de LeDock.

Hay que tener en cuenta que las posiciones del ferulenoles en 6aje_SinCof.csv son 706 y 909 (la que fue descargada tiene mejor posición) y no hubieran sido seleccionadas para DM.

Otro problema es que LeDock puede generar errores en el archivo “.pdb” como es el caso de C_89612265_01, sin embargo sus otras posiciones no lo tienen (Figura 6). En DRIVE > Videos > 01VID_C_89612265_01.mp4, se puede ver lo que este error genera en DM, esto es, una molécula irreal desprendida del medicamento dando vueltas en la solución.

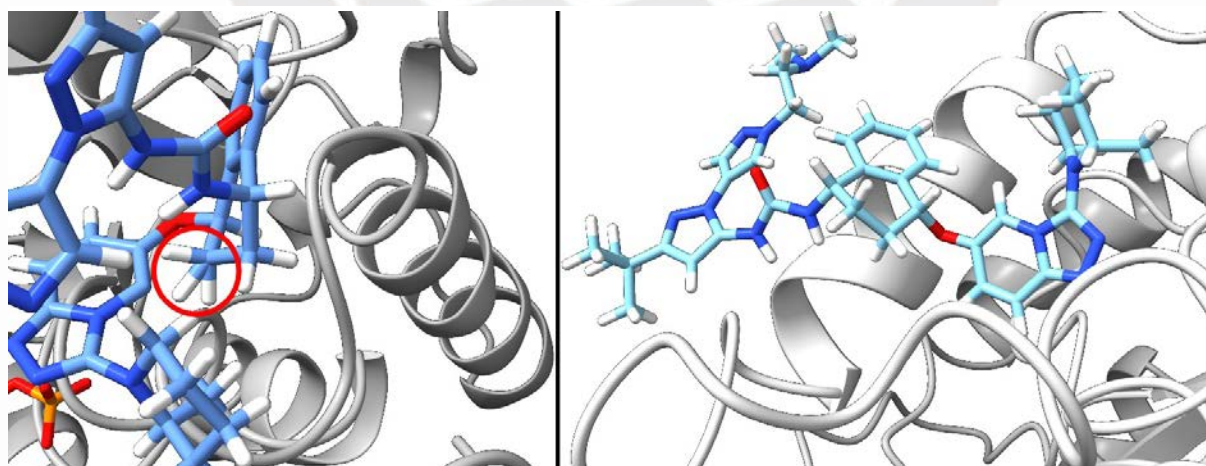


Figura 6 Error de AM. La imagen izquierda visualiza el “.pdb” generado por AM de la primera posición de C_89612265, el círculo rojo muestra el error (es decir, un carbono con cinco enlaces), mientras que la imagen de la derecha muestra la posición 16, la cual no presenta ese error.

En general, es necesario analizar las estructuras antes y después de las simulaciones de DM, para observar si se está usando y generando una estructura lógica. El siguiente ejemplo corresponde a C_89612265_16 (Figura 7). Las imágenes de todas las estructuras están en los dos primeros subcapítulos de “A.4.”.

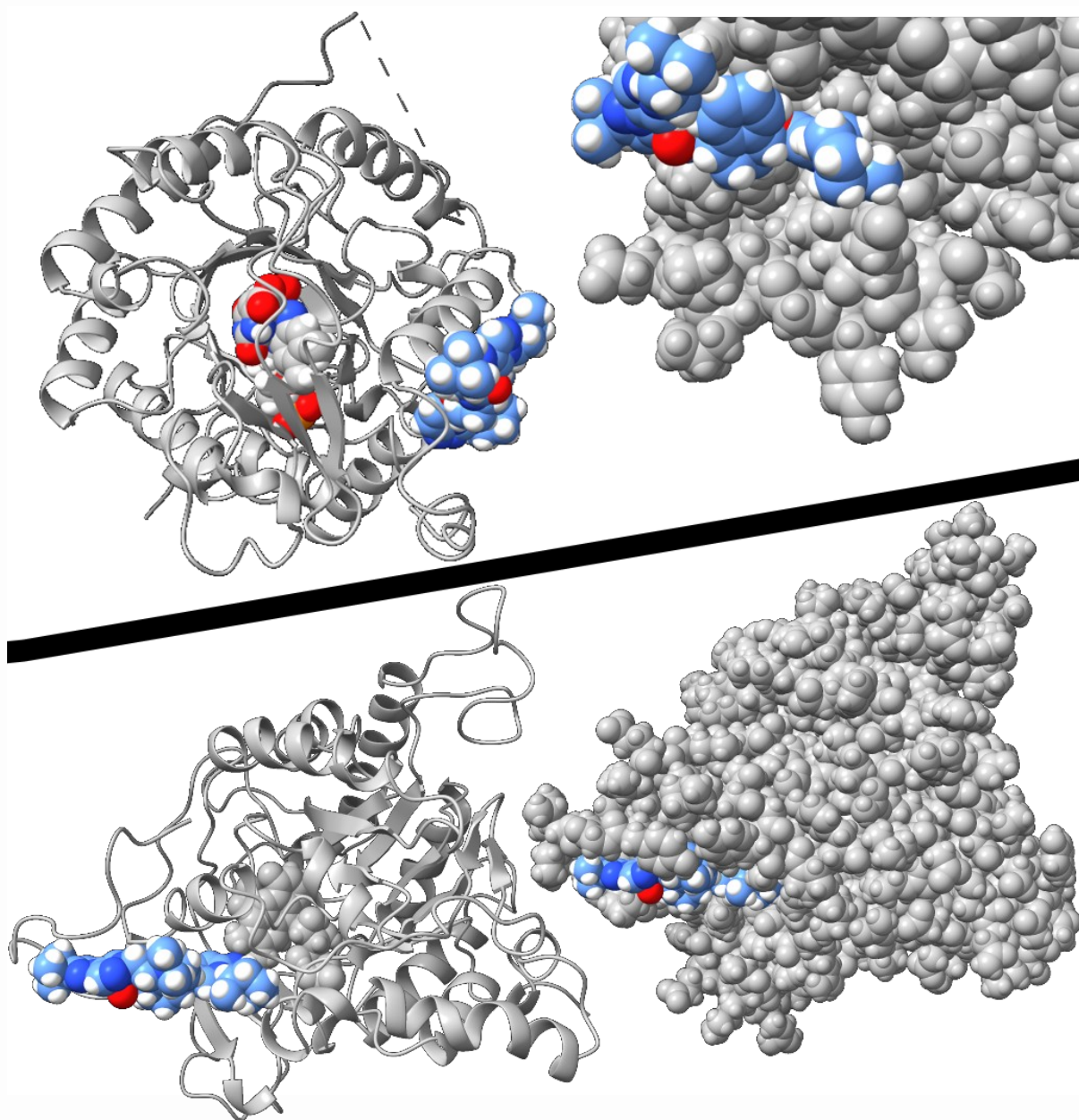


Figura 7 Antes y después de DM. Las imágenes corresponden a C_89612265_16: la superior se obtuvo del archivo “.pdb” de AM, mientras que la inferior, del último nano-segundo de la simulación de DM.

3.2. MMPBSA

Luego de obtener los resultados de DM, conviene revisar los valores de afinidad obtenidos. La Tabla 4 muestra estos valores para todas las 19 corridas, estas incluyen a las de moléculas iguales con diferentes conformaciones generadas por AM y a las tres del ferulenol. En otras palabras, se analizaron 14 medicamentos. Lo más resaltante es que el ferulenol tiene buenos valores de afinidad y que es superado sólo por algunos medicamentos. Sin embargo, también es importante tener en cuenta la estabilidad del complejo proteína-medicamento, la cual se intuye mediante los gráficos RMSD.

Tabla 4 Valores de afinidad: MMPBSA.

Nombre	Δ TOTAL(kcal/mol)
C_89612265_16	-54,41
A_5351322_01	-49,07
A_16004692_01	-47,47
Fd_54679300_01	-43,13
C_89612265_01	-42,09
F_54679300_05_ConCof	-40,28
C_4887_17	-39,17
F_54679300_05_sinCof	-37,31
C_4887_02	-33,55
B_42613186_01	-26,43
A_5353622_01	-22,4
B_3032325_01	-20,9
A_135564890_01	-20,63
A_9571836_01	-19,53
B_6133_01	-13,01
B_42613186_02	-10,75
B_134019_01	-10,4
B_5702160_01	-8,67
A_5394_01	0,47

3.3. RMSD

La Figura 8 muestra que la estabilidad es mayor cuando hay co-factores en el complejo: la variación tiende a una línea horizontal, mientras que “Sólo con ferulenol” está más alejado de la proteína y fluctúa más. Mientras que la Figura 9 muestra los gráficos RMSD más representativos. Todos los gráficos de este tipo están en “A.4.3.”.

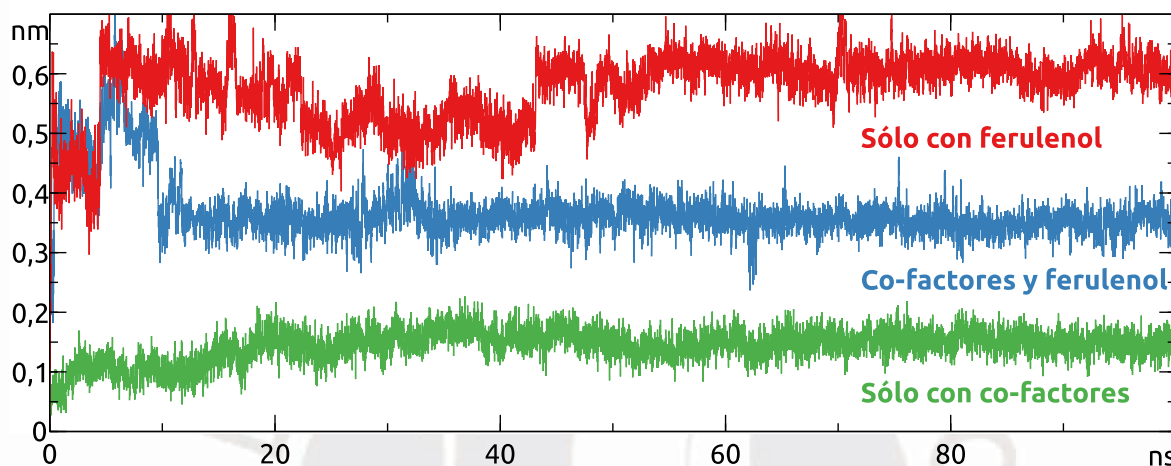


Figura 8 RMSD: ferulenol y co-factores. Espectro verde está más abajo porque usa un co-factor (este es ORO) como referencia un lugar del inhibidor (es decir, ferulenol): el co-factor esta dentro de la molécula, por lo cual su distancia es menor.

Sobre la Figura 9, el RMSD del C_89612265_01 presenta una gran fluctuación. Esta se debe al error que generó la estructura producida por AM: el pedazo libre del medicamento altera la distancia relativa. Respecto al C_89612265_16, al A_5351322_01 y al A_16004692_01, estos tienen buenos valores MMPBSA (es decir, -54,41, -49,07 y -47,47, respectivamente) pero sólo los dos primeros son estables. En contraste, el B_3032325_01 se estabiliza, pero su valor MMPBSA es malo, este es -20,9. Por lo tanto, los dos mencionados al último son malos candidatos.

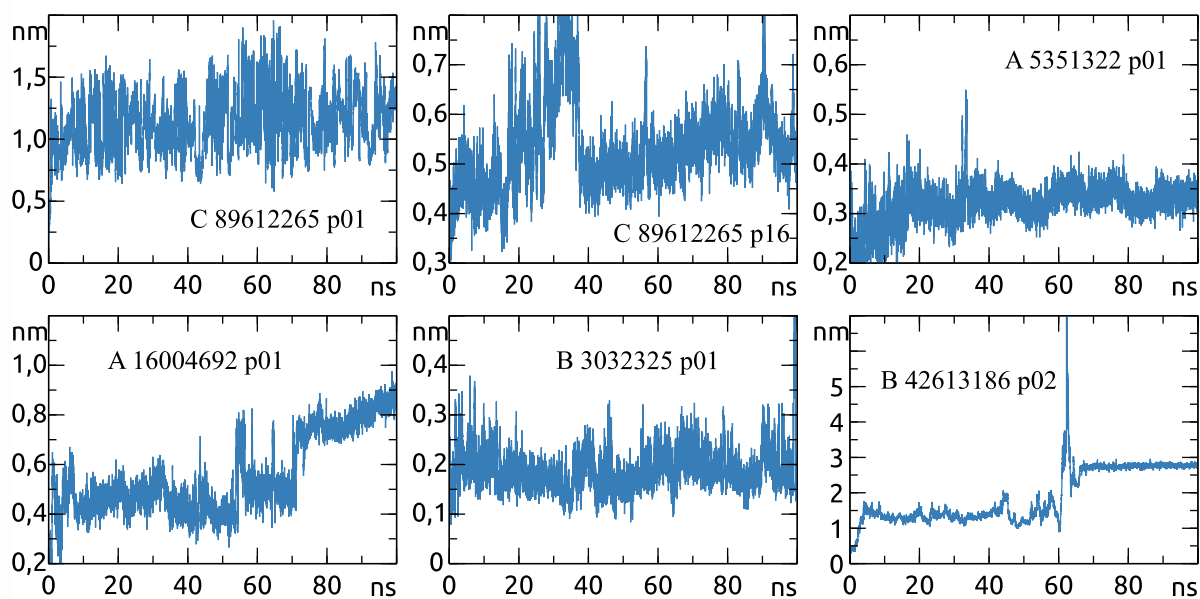


Figura 9 RMSD representativos. El “eje x” tiene la misma escala de 100ns pero hay que tener en cuenta que el “eje y” tiene diferentes rangos.

Por último, el B_42613186_02 no es un buen candidato (MMPBSA: -10,75), pero muestra la capacidad de GROMACS de simular como una molécula se desprende de una proteína y, luego, interactúa con esta en otra región. Lo cual genera que su RMSD se estabilice en una región superior, ya que la distancia promedio cambia, y tenga un pico bien pronunciado, el cual corresponde a la molécula libre. Sin embargo, para llegar a esta conclusión, debido a que los picos pueden corresponder a otros cambios estructurales, se tuvo que observar lo que ocurría en ese intervalo de tiempo. Lo cual se puede ver en DRIVE > Videos > 02VID_B_42613186_02_del59a166ns.mp4.

3.4. RMSF

Por último, los gráficos RMSF permiten sospechar de un efecto inhibitor. Sin embargo, incluyendo al ferulenol, no se observan cambios importantes del comportamiento de la proteína, exepcto para un caso. Lo cual indica que el efecto inhibitor se debe a una obstrucción del sitio activo más que a un cambio de conformación en la proteína que la haga inservible. La excepción es C_89612265_16, lo cual lo hace un mejor candidato aún (Figura 10 y “A.4.4.”).

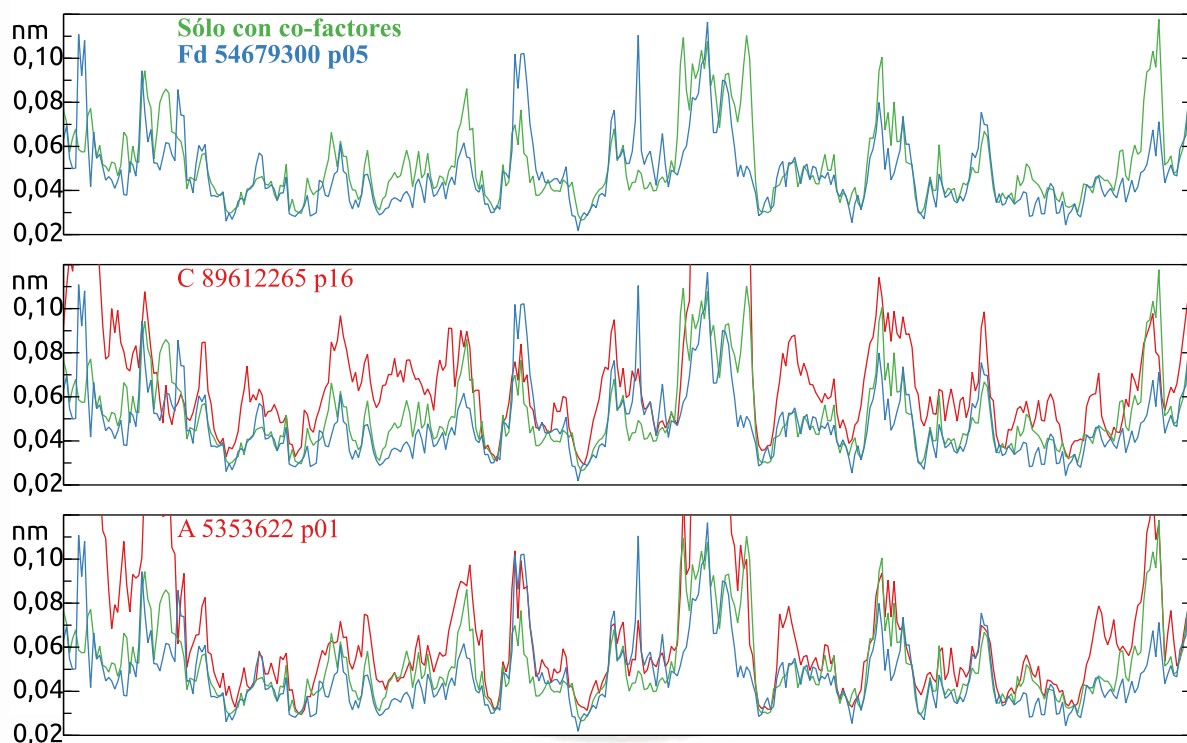


Figura 10 RMSF. Los tres gráficos corresponden (de arriba hacia abajo), primero, al de proteína sin inhibidor superpuesto con la proteína con ferulenol, al de C_89612265_16 y al de A_5351322_01 (estos dos últimos superpuestos, cada uno, con el primero).

Capítulo 4

4. Discusión y recomendaciones

Hay que tener en cuenta que en las simulaciones computacionales no está la membrana mitocondrial en la que la proteína “6AJE” se encuentra. Lo mismo ocurre con lo realizado *in vitro* por Sato et al., 2020 que mide una buena inhibición causada por el ferulenol en un purificado de “6AJE”, es decir, proteínas sin la membrana. Lo interesante es que los resultados concuerdan: el ferulenol genera inhibición *in vitro* y el valor bajo de MMPBSA (es decir, -43,13) la intuye. En otras palabras, en ambos casos no se incluye a la membrana. Por lo tanto, la afinidad que muestran podría no corresponder a la realidad. Una alternativa de mejora es incluir la membrana, pero eventualmente van a ser necesarias pruebas *in vivo* en el pollo.

Por otro lado, respecto al ferulenol, hay una gran diferencia entre los valores de afinidad MMPBSA y los de AM. Lo cual indica que LeDock no es bueno en detectar todos los mejores candidatos. Además, al correr los mismos medicamentos en proteínas diferentes (es decir, 6AJE y 5E34) se pueden filtrar medicamentos a los que LeDock les da mejor puntaje porque tiene sesgo hacia ellos, aunque también se eliminarían los que tienen actividad contra ambas proteínas, lo cual es bueno si el medicamento no es específico. Pero convendría usar una tercera proteína para complementar esa información.

La ventaja de LeDock es su rapidez y su libre acceso, pero se podrían desarrollar mejores programas con estas características. Por ejemplo, GWOVina, gracias a su algoritmo de búsqueda, tiene una mejor tasa de éxito (62 vs 29%) y una mayor rapidez (9 vs 10 segundos) que LeDock y otros programas. Además, puede darle cierta flexibilidad a las cadenas laterales (K. M. Wong et al., 2021).

En resumen, es necesario mejorar la tasa de éxito y el uso de recursos computacionales, es

decir, la rapidez de procesamiento. Sin embargo, mejorar uno suele perjudicar al otro. Por ejemplo, aumentar la flexibilidad de la proteína genera mejores resultados pero aumenta el uso de recursos computacionales, mientras que un mejor algoritmo de búsqueda beneficia a ambos. Por lo tanto, se necesitan mejores programas y computadoras más potentes. Lo cual también es aplicable a la DM, en particular simular la membrana requiere de mayores recursos pero tiene el potencial de dar mejores resultados.

Respecto a la selección de 6AJE, la principal razón fue la disponibilidad de la estructura y lo mencionado por Sato et al., 2020: el parásito necesita esa proteína para sobrevivir en su etapa intracelular. Sin embargo, convendría analizar otras proteínas, sobre todo porque se podrían atacar dos al mismo tiempo: esto puede dificultar la generación de resistencia.

En lo que corresponde a los medicamentos analizados, se obtuvieron dos buenos candidatos (Figura 11). El mejor es el C_89612265_16, ya que tiene el mejor valor de afinidad y deforma a la proteína. Este medicamento fue usado en un estudio para tratar la “Enfermedad pulmonar obstructiva crónica” (Mair & Singh, 2020; PubChem, 2024b). El otro candidato es el A_5351322_01, aunque este puede que no sea muy específico, ya que en AM también está entre los primeros para 5E34. Este medicamento corresponde al bisantreno, el cual es usado para tratar al cáncer, ya que inhibe la replicación del ADN (PubChem, 2024a). Sin embargo, respecto a estos dos candidatos, convendría analizar su costo de producción, investigar su capacidad de llegar a la proteína del parásito y, de ser necesario, desarrollar el medicamento para aumentar esta capacidad, y, luego, realizar estudios *in vivo* en el pollo.

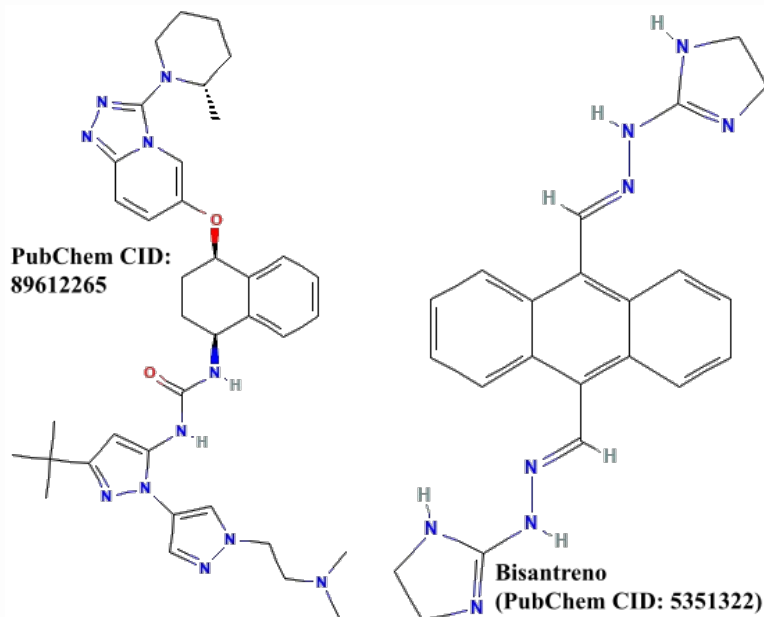


Figura 11 Candidatos encontrados. Estructura de los dos medicamentos más prometedores según los resultados obtenidos. Estructuras de PubChem, 2024b, 2024a.

Respecto a la tesis en general, esta cumple la función de un estudio exploratorio en el uso de técnicas computacionales para encontrar medicamentos que traten enfermedades. Sin embargo, cabe añadir que, de las técnicas computacionales utilizadas, LeDock no funciona bien y se sugiere reemplazarla por otra de AM, por ejemplo, usar GWOVina o desarrollar una nueva. Mientras que GROMACS sí tiene un gran potencial, por ejemplo, puede simular un desprendimiento del medicamento y posterior interacción en otra región de la proteína. Aunque a esta técnica de DM le falta una mejor interfaz y un manual más didáctico, el cual enseñe diferentes posibilidades de su uso. Además, convendría crear un programa que una y automatice el uso de AM y DM.

En otras palabras, los siguientes pasos a realizar son desarrollar mejores programas para AM y DM en base a los ya existentes y en lo posible integrarlos y automatizarlos, en particular un mejor procedimiento para calcular la afinidad medicamento-proteína. Además, conviene desarrollar estrategias para simular la proteína en membrana o la adhesión del medicamento a aquella. Sin embargo, esto requiere de mayores recursos computacionales.

Otro aspecto a mejorar son las bases de datos, sobre todo lo relacionado al orden, a la

cantidad de moléculas, a la calidad de la información y a su disponibilidad. Idealmente, conviene tener una mayor variedad de proteínas, completamente dilucidadas, de cada especie del parásito para tener más puntos de ataque y buscar que el tratamiento afecte a todas las *Eimeria* spp. Además de que los medicamentos tengan mejor información complementaria, por ejemplo, posibles usos medicinales, posible eficacia, precio aproximado y propiedades fisicoquímicas.

Respecto a lo anterior, hay que tener en cuenta los beneficios de documentos y programas de libre acceso para el desarrollo científico, sobre todo en lo relacionado con enfermedades.

Cabe mencionar que la parte experimental se realizó con *laptops*: convendría usar computadoras con mayor potencia (por ejemplo, *workstations* del año o super-computadoras) y usar mejores GPUs. Sin embargo, la capacidad de usar computadoras no tan potentes para realizar estos experimentos es una ventaja que se puede utilizar.

Por último, el procedimiento usado en la parte experimental se puede ampliar para otras enfermedades, incluidas las huérfanas, y se puede modificar para otros usos. Por ejemplo, en lugar de usar una proteína para buscar buenos candidatos de varios medicamentos, se puede buscar la funcionalidad de una molécula recientemente aislada usando una base de datos de varias proteínas.

Referencias

- Abenoja, J., Cotto-Rosario, A., & O'Connor, R. (2021). Boromycin Has Potent Anti-*Toxoplasma* and Anti-*Cryptosporidium* Activity. *Antimicrobial Agents and Chemotherapy*, 65(4), e01278-20. <https://doi.org/10.1128/AAC.01278-20>
- Alaimo, S., & Pulvirenti, A. (2019). Network-Based Drug Repositioning: Approaches, Resources, and Research Directions. En Q. Vanhaelen (Ed.), *Computational Methods for Drug Repurposing* (pp. 97-113). Springer. https://doi.org/10.1007/978-1-4939-8955-3_6
- Aldeghi, M., Bodkin, M. J., Knapp, S., & Biggin, P. C. (2017). Statistical Analysis on the Performance of Molecular Mechanics Poisson–Boltzmann Surface Area versus Absolute Binding Free Energy Calculations: Bromodomains as a Case Study. *Journal of Chemical Information and Modeling*, 57(9), 2203-2221. <https://doi.org/10.1021/acs.jcim.7b00347>
- AMBER. (2022). *Download Amber MD*. <https://ambermd.org/GetAmber.php>
- Bennett, C. E., Thomas, R., Williams, M., Zalasiewicz, J., Edgeworth, M., Miller, H., Coles, B., Foster, A., Burton, E. J., & Marume, U. (2018). The broiler chicken as a signal of a human reconfigured biosphere. *Royal Society Open Science*, 5(12), 180325. <https://doi.org/10.1098/rsos.180325>
- Bergeron, S., Pouliot, E., & Doyon, M. (2020). Commercial Poultry Production Stocking Density Influence on Bird Health and Performance Indicators. *Animals*, 10(8), 1253. <https://doi.org/10.3390/ani10081253>
- Blake, D. P., Knox, J., Dehaeck, B., Huntington, B., Rathinam, T., Ravipati, V., Ayoade, S., Gilbert, W., Adebambo, A. O., Jatau, I. D., Raman, M., Parker, D., Rushton, J., & Tomley, F. M. (2020). Re-calculating the cost of coccidiosis in chickens. *Veterinary Research*, 51(1), 115. <https://doi.org/10.1186/s13567-020-00837-2>
- Burleson, M. (2019). Poultry and other avian species, Broiler Industry Report. *Proceedings One hundred and twenty second annual meeting of the United States Animal Health Association, October 18-24, 2018*, 320-326. <https://www.usaha.org/proceedings>
- Chapman, H. D. (2014). Milestones in avian coccidiosis research: A review. *Poultry Science*, 93(3), 501-511. <https://doi.org/10.3382/ps.2013-03634>
- Chapman, H. D., Jeffers, T. K., & Williams, R. B. (2010). Forty years of monensin for the control of coccidiosis in poultry. *Poultry Science*, 89(9), 1788-1801. <https://doi.org/10.3382/ps.2010-00931>
- Ciani, O., & Jommi, C. (2014). The role of health technology assessment bodies in shaping drug development. *Drug Design, Development and Therapy*, 2273. <https://doi.org/10.2147/DDDT.S49935>
- Cobanoglu, F., Kucukyilmaz, K., Cinar, M., Bozkurt, M., Catli, A., & Bintas, E. (2014). Comparing the profitability of organic and conventional broiler production. *Revista Brasileira de Ciência Avícola*, 16(4), 403-410. <https://doi.org/10.1590/1516-635x1604403-410>

- Collett, S. R., Smith, J. A., Boulianne, M., Owen, R. L., Gingerich, E., Singer, R. S., Johnson, T. J., Hofacre, C. L., Berghaus, R. D., & Stewart-Brown, B. (2020). Principles of Disease Prevention, Diagnosis, and Control. En D. E. Swayne, M. Boulianne, C. M. Logue, L. R. McDougald, V. Nair, D. L. Suarez, S. Wit, T. Grimes, D. Johnson, M. Kromm, T. Y. Prajitno, I. Rubinoff, & G. Zavala (Eds.), *Diseases of Poultry* (1.^a ed., pp. 1-78). Wiley. <https://doi.org/10.1002/9781119371199.ch1>
- CONDA. (2021). *Installing on Linux—Conda 23.3.0.post13+db38e958b documentation*. <https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html>
- Exchange Rates UK. (2021). *British Pound to US Dollar Spot Exchange Rates for 2016*. <https://www.exchangerates.org.uk/GBP-USD-spot-exchange-rates-history-2016.html>
- FAOSTAT. (2023a). *Producer Prices*. <https://www.fao.org/faostat/en/#data/PP>
- FAOSTAT. (2023b). *Crops and livestock products*. <https://www.fao.org/faostat/en/#data/QCL>
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37. <https://doi.org/10.1609/aimag.v17i3.1230>
- FDA. (2022). New Drug Development and Review Process. *Center for Drug Evaluation and Research*. <https://www.fda.gov/drugs/cder-small-business-industry-assistance-sbia/new-drug-development-and-review-process>
- Fry, J. P., Mailloux, N. A., Love, D. C., Milli, M. C., & Cao, L. (2018). Feed conversion efficiency in aquaculture: Do we measure it correctly? *Environmental Research Letters*, 13(2), 024017. <https://doi.org/10.1088/1748-9326/aaa273>
- Fulton, R. M. (2020). Toxins and Poisons. En D. E. Swayne, M. Boulianne, C. M. Logue, L. R. McDougald, V. Nair, D. L. Suarez, S. Wit, T. Grimes, D. Johnson, M. Kromm, T. Y. Prajitno, I. Rubinoff, & G. Zavala (Eds.), *Diseases of Poultry* (1.^a ed., pp. 1349-1382). Wiley. <https://doi.org/10.1002/9781119371199.ch32>
- GitHub. (2022). *Releases · intel/compute-runtime*. <https://github.com/intel/compute-runtime/releases>
- Grinnan, D., Trankle, C., Andruska, A., Bloom, B., & Spiekerkoetter, E. (2019). Drug repositioning in pulmonary arterial hypertension: Challenges and opportunities. *Pulmonary Circulation*, 9(1), 1-18. <https://doi.org/10.1177/2045894019832226>
- GROMACS. (2023). *GROMACS 2023 documentation*. <https://manual.gromacs.org/current/index.html>
- Gustin, S., & McCarter, S. (2021). Poultry and other avian species, Association of Veterinarians in Broiler Production (AVBP) Current Diseases of Concern. *Proceedings One hundred and twenty third annual meeting of the United States Animal Health Association, October 24-30, 2019*, 304-309. <https://www.usaha.org/proceedings>
- Huang, F., Zhang, C., Liu, Q., Zhao, Y., Zhang, Y., Qin, Y., Li, X., Li, C., Zhou, C., Jin, N., & Jiang, C. (2020). Identification of amitriptyline HCl, flavin adenine dinucleotide, azacitidine and calcitriol as repurposing drugs for influenza A H5N1 virus-induced

- lung injury. *PLOS Pathogens*, 16(3), e1008341. <https://doi.org/10.1371/journal.ppat.1008341>
- Hughes, J., Rees, S., Kalindjian, S., & Philpott, K. (2011). Principles of early drug discovery: Principles of early drug discovery. *British Journal of Pharmacology*, 162(6), 1239-1249. <https://doi.org/10.1111/j.1476-5381.2010.01127.x>
- Ibrahim, D., El-Sayed, R., Khater, S. I., Said, E. N., & El-Mandrawy, S. A. M. (2018). Changing dietary n-6:n-3 ratio using different oil sources affects performance, behavior, cytokines mRNA expression and meat fatty acid profile of broiler chickens. *Animal Nutrition*, 4(1), 44-51. <https://doi.org/10.1016/j.aninu.2017.08.003>
- INEI. (2022). *Peru Compendio Estadístico 2022: Tomo 2*. Instituto Nacional de Estadística e Informática. <https://www.gob.pe/institucion/inei/informes-publicaciones/3655985-compendio-estadistico-peru-2022>
- Intel. (2022a). *Intel Core i3-1005G1 Processor*. <https://www.intel.com/content/www/us/en/products/sku/196588/intel-core-i31005g1-processor-4m-cache-up-to-3-40-ghz/specifications.html?wapkw=i3-1005G1>
- Intel. (2022b). *Intel Core i5-2430M Processor*. <https://www.intel.com/content/www/us/en/products/sku/53450/intel-core-i52430m-processor-3m-cache-up-to-3-00-ghz/specifications.html?wapkw=i5-2430M>
- Intel. (2022c). *Intel Core i7-4710HQ Processor*. <https://www.intel.com/content/www/us/en/products/sku/78930/intel-core-i74710hq-processor-6m-cache-up-to-3-50-ghz/specifications.html?wapkw=i7-4710HQ>
- Jayawardene, K. L. T. D., Palombo, E. A., & Boag, P. R. (2021). Natural Products Are a Promising Source for Anthelmintic Drug Discovery. *Biomolecules*, 11(10), 1457. <https://doi.org/10.3390/biom11101457>
- Johnson, D. (2017). Transmissible diseases of poultry and other avian species, Broiler Industry Report. *Proceedings One hundred and twentieth annual meeting of the United States Animal Health Association, October 13–19, 2016*, 375-380. <https://www.usaha.org/proceedings>
- Lemkul, J. A. (2018). *GROMACS Tutorials*. <http://www.mdtutorials.com/gmx/>
- Li, Y., Xing, S., Wang, X., Li, X., Zhang, M., & Feng, J. (2021). Effects of Increasing Stocking Density on the Performance and Ileal Microbiota of Broilers. *The Journal of Poultry Science*, 59(3), 291-296. <https://doi.org/10.2141/jpsa.0210089>
- Mair, S., & Singh, D. (2020). *A Study to Investigate Safety, Tolerability, Pharmacokinetics and Pharmacodynamics of Single and Repeat Doses of CHF6297 in Healthy Subjects and Patients With COPD*. <https://clinicaltrials.gov/study/NCT02815488>
- McDougald, L. R., Cervantes, H. M., Jenkins, M. C., Hess, M., & Beckstead, R. (2020). Protozoal Infections. En D. E. Swayne, M. Boulianne, C. M. Logue, L. R. McDougald, V. Nair, D. L. Suarez, S. Wit, T. Grimes, D. Johnson, M. Kromm, T. Y. Prajitno, I. Rubinoff, & G. Zavala (Eds.), *Diseases of Poultry* (1.^a ed., pp. 1192-1254). Wiley. <https://doi.org/10.1002/9781119371199.ch28>

- Mountainspring, G., & Burleson, M. (2018). Poultry and other avian species, Broiler Industry Report. *Proceedings One hundred and twenty first annual meeting of the United States Animal Health Association, October 12–18, 2017*, 383-387.
<https://www.usaha.org/proceedings>
- Mueller, S., Taddei, L., Albiker, D., Kreuzer, M., Siegrist, M., Messikommer, R. E., & Gangnat, I. D. M. (2020). Growth, carcass, and meat quality of 2 dual-purpose chickens and a layer hybrid grown for 67 or 84 D compared with slow-growing broilers. *Journal of Applied Poultry Research*, 29(1), 185-196.
<https://doi.org/10.1016/j.japr.2019.10.005>
- Muir, W. M., Wong, G. K.-S., Zhang, Y., Wang, J., Groenen, M. A. M., Crooijmans, R. P. M. A., Megens, H.-J., Zhang, H., Okimoto, R., Vereijken, A., Jungerius, A., Albers, G. A. A., Lawley, C. T., Delany, M. E., MacEachern, S., & Cheng, H. H. (2008). Genome-wide assessment of worldwide chicken SNP genetic diversity indicates significant absence of rare alleles in commercial breeds. *Proceedings of the National Academy of Sciences*, 105(45), 17312-17317.
<https://doi.org/10.1073/pnas.0806569105>
- Noack, S., Chapman, H. D., & Selzer, P. M. (2019). Anticoccidial drugs of the livestock industry. *Parasitology Research*, 118(7), 2009-2026. <https://doi.org/10.1007/s00436-019-06343-5>
- NVIDIA. (2022). *CUDA Toolkit 11.7 Downloads*. NVIDIA Developer.
<https://developer.nvidia.com/cuda-downloads>
- Orkusz, A. (2021). Edible Insects versus Meat—Nutritional Comparison: Knowledge of Their Composition Is the Key to Good Health. *Nutrients*, 13(4), 1207.
<https://doi.org/10.3390/nu13041207>
- Panic, G., Duthaler, U., Speich, B., & Keiser, J. (2014). Repurposing drugs for the treatment and control of helminth infections. *International Journal for Parasitology: Drugs and Drug Resistance*, 4(3), 185-200. <https://doi.org/10.1016/j.ijpddr.2014.07.002>
- Parker, A., Lawson, M. A. E., Vaux, L., & Pin, C. (2018). Host-microbe interaction in the gastrointestinal tract. *Environmental Microbiology*, 20(7), 2337-2353.
<https://doi.org/10.1111/1462-2920.13926>
- PubChem. (2024a). *Bisantrene*. <https://pubchem.ncbi.nlm.nih.gov/compound/5351322>
- PubChem. (2024b). *Urea, N-(1'-(2-(dimethylamino)ethyl)-3-(1,1-dimethylethyl)(1,4'-bi-1H-pyrazol)-5-yl)-N'-((1S,4R)-1,2,3,4-tetrahydro-4-((3-((2S)-2-methyl-1-piperidinyl)-1,2,4-triazolo(4,3-a)pyridin-6-yl)oxy)-1-naphthalenyl)-*.
<https://pubchem.ncbi.nlm.nih.gov/compound/89612265>
- RCSB Protein Data Bank. (2015). *RCSB PDB - 5E34: Crystal structure of H5 hemagglutinin mutant (N224K, Q226L, N158D and L133a deletion) from the influenza virus A/chicken/Vietnam/NCVD-093/2008 (H5N1) with LSTa*.
<https://www.rcsb.org/structure/5E34>
- RCSB Protein Data Bank. (2019). *RCSB PDB - 6AJE: Crystal structure of DHODH in complex with ferulenol from Eimeria tenella*. <https://www.rcsb.org/structure/6AJE>

- Reis, A. C., Kolvenbach, B. A., Nunes, O. C., & Corvini, P. F. X. (2020a). Biodegradation of antibiotics: The new resistance determinants – part I. *New Biotechnology*, *54*, 34-51. <https://doi.org/10.1016/j.nbt.2019.08.002>
- Reis, A. C., Kolvenbach, B. A., Nunes, O. C., & Corvini, P. F. X. (2020b). Biodegradation of antibiotics: The new resistance determinants – part II. *New Biotechnology*, *54*, 13-27. <https://doi.org/10.1016/j.nbt.2019.08.003>
- Russow, L.-M. (1999). Bioethics, Animal Research, and Ethical Theory. *ILAR Journal*, *40*(1), 15-21. <https://doi.org/10.1093/ilar.40.1.15>
- Salmaso, V., & Moro, S. (2018). Bridging Molecular Docking to Molecular Dynamics in Exploring Ligand-Protein Recognition Process: An Overview. *Frontiers in Pharmacology*, *9*, 923. <https://doi.org/10.3389/fphar.2018.00923>
- Sato, D., Hartuti, E. D., Inaoka, D. K., Sakura, T., Amalia, E., Nagahama, M., Yoshioka, Y., Tsuji, N., Nozaki, T., Kita, K., Harada, S., Matsubayashi, M., & Shiba, T. (2020). Structural and Biochemical Features of Eimeria tenella Dihydroorotate Dehydrogenase, a Potential Drug Target. *Genes*, *11*(12), 1468. <https://doi.org/10.3390/genes11121468>
- SEIKAGAKU. (2021). *Process of new drug development*. <https://www.seikagaku.co.jp/en/development/flow.html>
- Silva, A. (2022). *Tutorial Using ACPYPE for GROMACS*. GitHub. <https://github.com/alanwilter/acpype/wiki/Tutorial-Using-ACPYPE-for-GROMACS>
- Singh, V. K., & Seed, T. M. (2021). How necessary are animal models for modern drug discovery? *Expert Opinion on Drug Discovery*, *16*(12), 1391-1397. <https://doi.org/10.1080/17460441.2021.1972255>
- Sohraby, F., Bagheri, M., & Aryapour, H. (2019). Performing an In Silico Repurposing of Existing Drugs by Combining Virtual Screening and Molecular Dynamics Simulation. En Q. Vanhaelen (Ed.), *Computational Methods for Drug Repurposing* (Vol. 1903, pp. 23-43). Springer New York. https://doi.org/10.1007/978-1-4939-8955-3_2
- Strobel, C., Jahreis, G., & Kuhnt, K. (2012). Survey of n-3 and n-6 polyunsaturated fatty acids in fish and fish products. *Lipids in Health and Disease*, *11*(1), 144. <https://doi.org/10.1186/1476-511X-11-144>
- Valdes-Tresanco, M. S., Valdes-Tresanco, M. E., Valiente, P. A., & Moreno Frías, E. (2022). *gmx_MMPBSA_ana—gmx_MMPBSA Documentation*. https://valdes-tresanco-ms.github.io/gmx_MMPBSA/dev/analyzer/
- Valdés-Tresanco, M. S., Valdés-Tresanco, M. E., Valiente, P. A., & Moreno Frías, E. (2023). *gmx-MMPBSA: gmx_MMPBSA is a new tool based on AMBER's MMPBSA.py aiming to perform end-state free energy calculations with GROMACS files* (Versión 1.6.0) [Software]. <https://pypi.org/project/gmx-MMPBSA/>
- van de Waterbeemd, H., Carter, R. E., Grassy, G., Kubinyi, H., Martin, Y. C., Tute, M. S., & Willett, P. (1997). Glossary of terms used in computational drug design (IUPAC Recommendations 1997). *Pure and Applied Chemistry*, *69*(5), 1137-1152. <https://doi.org/10.1351/pac199769051137>

- Webster, I. (2023). *U.S. Inflation Calculator: 1635→2023, Department of Labor data*. <https://www.officialdata.org/>
- Wikimedia. (2007). *Apicomplexan life cycle: 1-zygote, 2-sporozoites, 3-merozoites, 4-gametes*. [Graphic]. Autor: Franciscosp2. https://commons.wikimedia.org/wiki/File:Apicomplexa_life_cycle_v2.svg
- Williams, R. B. (1999). A compartmentalised model for the estimation of the cost of coccidiosis to the world's chicken production industry. *International Journal for Parasitology*, 29(8), 1209-1229. [https://doi.org/10.1016/S0020-7519\(99\)00086-7](https://doi.org/10.1016/S0020-7519(99)00086-7)
- Wong, A. (2019). Unknown Risk on the Farm: Does Agricultural Use of Ionophores Contribute to the Burden of Antimicrobial Resistance? *mSphere*, 4(5), e00433-19. <https://doi.org/10.1128/mSphere.00433-19>
- Wong, K. M., Tai, H. K., & Siu, S. W. I. (2021). GWOVina: A grey wolf optimization approach to rigid and flexible receptor docking. *Chemical Biology & Drug Design*, 97(1), 97-110. <https://doi.org/10.1111/cbdd.13764>
- Woods, D. J., Vaillancourt, V. A., Wendt, J. A., & Meeus, P. F. (2011). Discovery and development of veterinary antiparasitic drugs: Past, present and future. *Future Medicinal Chemistry*, 3(7), 887-896. <https://doi.org/10.4155/fmc.11.39>
- XavierB, alevilla, & Marwa_92. (2021). *Protein/Ligand complex amber production question*. GROMACS Forums. <https://gromacs.bioexcel.eu/t/protein-ligand-complex-amber-production-question/1247>
- Ximing XU. (2016). *Molecular docking tutorial—Ledock* [Video recording]. <https://www.youtube.com/watch?v=mKTKmoGgl8o>
- Yin, W., Xu, T., Altai, M., Oroujeni, M., Zhang, J., Vorobyeva, A., Vorontsova, O., Vtorushin, S. V., Tolmachev, V., Gräslund, T., & Orlova, A. (2021). The Influence of Domain Permutations of an Albumin-Binding Domain-Fused HER2-Targeting Affibody-Based Drug Conjugate on Tumor Cell Proliferation and Therapy Efficacy. *Pharmaceutics*, 13(11), 1974. <https://doi.org/10.3390/pharmaceutics13111974>

Anexos

En esta parte hay comandos (en Bash) que se usaron en la Terminal. Para poder identificarlos, se colocó “\$ ” adelante de estos (excluir al copiar). Cabe mencionar que se puede abrir la Terminal en una carpeta específica: clic izquierdo en la carpeta y seleccionar la opción adecuada; o correr el comando “\$ cd /DIRECCIÓN_DE_LA_CARPETA”. Además, en el caso que este archivo esté en PDF, sus comandos se pueden copiar mal debido a caracteres que cortan las líneas (por ejemplo, %, \$, -). Por lo tanto, en el DRIVE, se encuentra este archivo en “.odt” y la carpeta “Comandos”, en la cual hay archivos de texto plano que contienen a los comandos utilizados.

A.1. Programas y computadoras

Primero, se menciona a las características de las computadoras utilizadas y, luego, a los programas que se les instalaron.

A.1.1. Computadoras

Se utilizaron tres *laptops*: “i3”, “i5” e “i7”. Sus características principales se visualizaron con “\$ cat /proc/cpuinfo;lspci;cat /etc/os-release”, lo obtenido se usó en las siguientes dos tablas. La “Tabla 5” da información sobre las unidades centrales de procesamiento (CPUs), las cuales tienen GPUs integradas. Sin embargo, sólo la computadora “i7” tiene adicionalmente una GPU dedicada (esta es NVIDIA GeForce GTX 860M).

Tabla 5 CPUs de las *laptops*. Complementada con Intel, 2022a, 2022b, 2022c.

	Intel(R) Core(TM)	Núcleos (hilos)	Frecuencia máxima (GHz)	(Trimestre y) año de lanzamiento	GPU integrada
“i3”	i3-1005G1	2 (4)	3.4	(Q3) 2019	UHD Graphics for 10th Gen
“i5”	i5-2430M	2 (4)	3	(Q4) 2011	HD Graphics 3000
“i7”	i7-4710HQ	4 (8)	3.5	(Q2) 2014	HD Graphics 4600

Asimismo, como se ve en la Tabla 6, “i7” es más rápida que las otras dos computadoras en realizar simulaciones de DM mediante GROMACS, esta parte es la más lenta del proceso. Cabe mencionar que “ns/d” se refiere a los nano-segundos de simulación generados entre los días que demora producirlos. Además, los valores de esta tabla se obtuvieron de pocas simulaciones del mismo sistema y sólo en el caso de “i3” fueron discontinuas. También, una interfaz gráfica de usuario (GUI) consume recursos, por lo tanto, “i5” e “i7” no la tienen al no necesitarla.

Tabla 6 Desempeño de las *laptops*.

	Sistema operativo	Programas	GPU que utiliza GROMACS	Velocidad de GROMACS (ns/d)
“i3”	Ubuntu 22.04.1 LTS	Todos	La integrada	5.9 a 7.5
“i5”	Debian 11 (sin GUI)	GROMACS	No la detecta	3.9
“i7”			NVIDIA GeForce GTX 860M	24

A.1.2. Programas

Los principales programas son LeDock 1.0 y GROMACS 2022.3, pero estos requieren de otros programas que los complementan y permiten su uso. Estos son ChimeraX 1.4, LePro 1.0, ACPYPE 2022.7.21, gmx_mmpbsa 1.5.6, OpenBabel 3.1.1 y los programas que permiten las instalaciones, por ejemplo, Miniconda3 y cmake.

ChimeraX se usa para la edición de las moléculas y la visualización de estas y de sus simulaciones. OpenBabel 3.1.1 se usa para cambiar los formatos de las moléculas y es requerido en parte del AM y de la DM. En la primera también se usan a LeDock y LePro, y, en la segunda, GROMACS, ACPYPE y gmx_mmpbsa.

Por un lado, Ximing XU, 2016 dio información para instalar “LePro” y “LeDock”, además, utiliza Visual Molecular Dynamics (VMD) como alternativa a ChimeraX.

Respecto a ACPYPE y gmx_mmpbsa, se utilizó información de AMBER, 2022; CONDA, 2021; Silva, 2022; Valdes-Tresanco et al., 2022; Valdés-Tresanco et al., 2023. Acerca de

GROMACS, se aconseja revisar GROMACS, 2023; Lemkul, 2018; XavierB et al., 2021, particularmente <http://www.mdtutorials.com/gmx/complex/index.html> (tutorial), <https://manual.gromacs.org/current/user-guide/cmdline.html> (comandos) y <https://manual.gromacs.org/current/user-guide/mdp-options.html> (parámetros de “.mdp”).

A.1.3. Instalación

Las instrucciones para la instalación de los programas se muestran a continuación en tres categorías. Las instalaciones se realizaron entre julio y agosto del 2022.

A) Visualización y edición de las moléculas

a) UCSF ChimeraX: version 1.4 (2022-06-03)

Primero, se descargó de <https://www.rbvi.ucsf.edu/chimerax/download.html> el archivo “ucsf-chimerax_1.4ubuntu20.04_amd64.deb”.

Finalmente, en la carpeta donde está el archivo, se corrió “\$ sudo dpkg -i ucsf-chimerax_1.4ubuntu20.04_amd64.deb”.

B) AM

a) LePro y LeDock: v1.0 Copyright 2013-21 (C) H. Zhao PhD

Primero, se descargó, de <http://www.lephar.com/download.htm>, “lepro_linux_x86” y “ledock_linux_x86”. Luego, donde están estos 2 archivos, se corren los siguientes comandos.

```
$ chmod +x *_linux_x86
$ sudo mv lepro_linux_x86 /usr/local/bin/lepro;sudo mv ledock_linux_x86
/usr/local/bin/ledock
```

Los programas “LePro” y “LeDock” se pueden correr con los comandos “\$ lepro” y “\$ ledock”, respectivamente (si se corren solos aparece su descripción).

b) OpenBabel 3.1.1

Su instalación está en la siguiente parte (es decir, DM), ya que también usa este programa.

C) DM

a) ACPYPE 2022.7.21, gmx_mmpbsa 1.5.6 y OpenBabel 3.1.1

Primero, se instaló Miniconda3. Para esto se descargó “Miniconda3 Linux 64-bit” (con “Python 3.9”) de <https://docs.conda.io/en/latest/miniconda.html#linux-installers>, y se corrió lo siguiente donde está el archivo.

```
$ bash Miniconda3-py39_4.12.0-Linux-x86_64.sh
$ . ~/.bashrc
$ conda update -n base -c defaults conda
$ conda config --set channel_priority strict
$ sed -i "1achannels:\n- conda-forge\n- anaconda\n- defaults" ~/.condarc
$ conda config --set auto_activate_base false
```

Luego, se instalaron los programas que se van a utilizar, en un ambiente de nombre “gromacs”, con los siguientes comandos.

```
$ conda create -n gromacs;conda activate gromacs
$ conda install mpi4py ambertools acpype compilers;pip install gmx-MMPBSA PyQt6
```

Para utilizar estos programas, se tiene que correr antes “\$ conda activate gromacs”. Mientras que “\$ conda list” se usa para ver programas instalados en el ambiente y “\$ conda update --all” para actualizarlos.

b) GROMACS 2022.3 (Revisar <https://manual.gromacs.org/current/install-guide/index.html>)

Primero, se corrieron los siguientes comandos. El tercero es opcional, aunque instala algunos paquetes que podrían ser útiles.

```
$ sudo apt update -y;sudo apt upgrade -y
$ [ 1 -eq 2 ];until [ $? -eq 0 ];do sudo apt -y install build-essential opencl-headers ocl-icd-
opencl-dev libhwloc-dev libblas-dev liblapack-dev libclfft-dev cmake pkg-config python3
python3-pip python3-mpi4py libopenmpi-dev;done
$ [ 1 -eq 2 ];until [ $? -eq 0 ];do sudo apt -y install g++-multilib gcc-multilib cmake-format
ninja-build python3-venv python3-numpy;done
```

En segundo lugar, se instalaron los respectivos Controladores de dispositivo (“drivers”). Para la computadora con tarjeta NVIDIA, se instaló CUDA. Mientras que para las que van a usar GPU Intel, se instaló “OpenCL GPU driver (NEO)”: i7 usa NVIDIA, mientras que i3 e

i5, Intel (ver “A.1.1.”). Aunque GROMACS no detectó la GPU de la i5. Para el respectivo caso se corre lo siguiente en una nueva carpeta que después de la instalación se puede borrar.

i) CUDA 11.7.1 (Se siguieron las instrucciones de NVIDIA, 2022 para “Linux x86_64 Debian 11 deb (local)”. Además, el comando “\$ nvidia-smi” va a servir para ver el funcionamiento del GPU NVIDIA.)

```
$ sudo apt -y install dkms software-properties-common nvidia-detect;nvidia-detect
$ wget https://developer.download.nvidia.com/compute/cuda/11.7.1/local_installers/cuda-repo-debian11-11-7-local_11.7.1-515.65.01-1_amd64.deb;sudo dpkg -i cuda-repo-debian11-11-7-local_11.7.1-515.65.01-1_amd64.deb;sudo cp /var/cuda-repo-debian11-11-7-local/cuda-*-keyring.gpg /usr/share/keyrings/;sudo add-apt-repository contrib;sudo apt update -y;sudo apt upgrade -y;sudo apt -y install cuda
$ sudo apt install -y nvidia
```

ii) OpenCL GPU driver (NEO). Los siguientes comandos se obtuvieron de GitHub, 2022, sin embargo, la pagina web de esta fuente debe de tener una versión más reciente.

```
$ wget https://github.com/intel/intel-graphics-compiler/releases/download/igc-1.0.12149.1/intel-igc-core_1.0.12149.1_amd64.deb;wget https://github.com/intel/intel-graphics-compiler/releases/download/igc-1.0.12149.1/intel-igc-openc1_1.0.12149.1_amd64.deb;wget https://github.com/intel/compute-runtime/releases/download/22.38.24278/intel-level-zero-gpu-dbgSYM_1.3.24278_amd64.ddeb;wget https://github.com/intel/compute-runtime/releases/download/22.38.24278/intel-level-zero-gpu_1.3.24278_amd64.deb;wget https://github.com/intel/compute-runtime/releases/download/22.38.24278/intel-openc1-icd-dbgSYM_22.38.24278_amd64.ddeb;wget https://github.com/intel/compute-runtime/releases/download/22.38.24278/intel-openc1-icd_22.38.24278_amd64.deb;wget https://github.com/intel/compute-runtime/releases/download/22.38.24278/libigdgmm12_22.1.8_amd64.deb
$ sudo dpkg -i *
```

Tercero, en una nueva carpeta, se descargó el archivo gromacs-2022.3.tar.gz de <https://doi.org/10.5281/zenodo.7037338>. Sin embargo, de <https://manual.gromacs.org/current/download.html> se pude obtener la última versión.

En la carpeta donde se descargó el archivo, se corrió lo siguiente para la computadora que utiliza NVIDIA. Para las que utilizan GPU Intel, se corre lo mismo, excepto que se cambia “-DGMX_GPU=CUDA” por “-DGMX_GPU=OpenCL”

```
DGMX_GPU_NB_CLUSTER_SIZE=4".
```

```
$ tar xzf gromacs-2022.3.tar.gz  
$ cd gromacs-2022.3;mkdir build;cd build  
$ (cmake .. -DGMX_BUILD_OWN_FFTW=ON -  
  DREGRESSIONTEST_DOWNLOAD=ON -DGMX_GPU=CUDA -  
  DGMX_HWLOC=ON -DCMAKE_INSTALL_PREFIX=~/.gromacs) 2>&1|tee -a t.txt  
$ (make;make check) 2>&1|tee -a t.txt  
$ sudo make install
```

En “t.txt”, se van a guardar los mensajes que aparecen en la Terminal al ejecutar el respectivo comando. Esto permite corregir errores. Por ejemplo, en los mensajes de “cmake” se puede buscar “failed” y “not”, y tratar de solucionar esto instalando paquetes que falten (y, luego, se borra el contenido de la carpeta “build” y se vuelve a correr “cmake”). En el caso de “make check”, por el final aparece “100% tests passed, 0 tests failed out of 88” (aunque si falla alguna prueba, puede que GROMACS también funcione).

Después de la instalación, se puede barrar la carpeta donde se colocó gromacs-2022.3.tar.gz. Además, para activar GROMACS, se tiene que correr “\$ source ~/.gromacs/bin/GMXRC”. Por otro lado, “\$ gmx -version” permite ver información del programa.

A.2. Información complementaria del procedimiento

A.2.1. Búsqueda en ChEMBL

Se utilizaron filtros en ChEMBL y comandos para que entren en la barra de búsqueda de PubChem.

Los filtros que se aplicaron fueron “Drugs”; 3 y 4 en “Max Phase” (Fase de la investigación clínica); y de 4 a 599 en “Molecular Weight” (Masa molecular). Se obtuvieron 2 941 (estas cantidades se van a reducir porque no todos sus medicamentos tienen una estructura molecular asociada o una estructura 3D disponible). Luego, se descargaron los medicamentos

en formato “.tsv”. Después, se abrió el archivo “.tsv” utilizando LibreOffice, se eliminaron los medicamentos que no tenían una estructura molecular asignada (es decir, los que no tenían “SMILES”), se copiaron los códigos, uno por línea, de la categoría “Parent Molecule”, en un archivo de texto plano (por ejemplo, 004a599.txt) (para crear este tipo de archivos se puede usar “\$ >NOMBRE.txt”) dentro de una carpeta vacía y se corrió lo siguiente en esta (el comando se puede usar para varios archivos “.txt”).

```
$ for f in *.txt;do split -l 250 --numeric=1 "$f" "${f%.txt}_";done;for f in !(*.txt);do sed -i 'a;N;$!ba;s/\n/ OR /g' $f;mv "$f" "${f}p.txt";done
```

Lo cual generó archivos con la estructura “[nombre de procedencia]_[número del grupo]p.txt”, por ejemplo, 004a599_02p.txt.

Estos archivos contienen los nombres de los medicamentos con “OR” entre cada uno y en grupos de hasta 250, por ejemplo, “CHEMBL1200691 OR CHEMBL470670 OR ...”. Lo cual permitió copiar el texto de cada archivo en el buscador de PubChem, para luego descargar los resultados en SDF 3D. Para un caso, apareció un mensaje de error al buscar en PubChem: se solucionó al buscar primero una parte y luego la restante. Además, al descargar los archivos “.sdf”, conviene cambiarles el nombre para tenerlos identificados.

A.2.2. Acoplamiento molecular

Primero, se trabajó con el archivo de la proteína (en este caso el 6aje.pdb). El cual se abrió en ChimeraX. Para abrir un archivo se utiliza “File/‘Open...’”, si se vuelve a utilizar para otro archivo (o el mismo) se van a abrir ambos. Para evitar esto, se usa “File/Close Session”. Mientras que para guardar la sesión, se usa “File/‘Save...’”.

Luego de abrir el archivo, se borran estructuras de ser necesario. Por ejemplo, en este caso, se conservó solamente la cadena A: se seleccionó con “Select/Chains/dhodehase/Chain A”, se

invirtió la selección con “Select/Invert” y esta se eliminó con “Actions/‘Atoms/Bonds’/Delete”. Ya que las 4 cadenas que contenía el archivo eran idénticas. Después, se agregan hidrógenos (esto es, usar “addh” en la barra de comandos de ChimeraX) y se guarda el archivo como 6aje_A.

Además, los co-factores y el inhibidor se pueden seleccionar en “Select/Residues”. En este caso se quieren usar los co-factores y el inhibidor para la simulación de DM. Por lo tanto, se abrió 6aje_A.pdb. Luego, se borró todo menos la molécula que se quiere y se guardó en formato “.pdb”. Después, se cerró la sesión. Finalmente, se volvió a abrir 6aje_A.pdb y se repitió el proceso para las otras moléculas. Con estos pasos se generaron los archivos cof_fmn.pdb, cof_oro.pdb y l_ferulenol.pdb, los cuales se guardaron en una nueva carpeta de nombre MD_6aje. En esta, para evitar posibles conflictos con ACPYPE, se corrió lo siguiente.

```
$ for f in *.pdb;do a=$(grep -n -m 1 '^HETATM' $f|cut -d : -f1-1);if (($a > 0));then sed -i "1,$a" d" $f;fi;done
```

Respecto a cómo realiza LeDock el AM, hay que tener en cuenta lo siguiente. (1) El archivo “.pdb” de la proteína utiliza “HETATM” para identificar a los co-factores y al inhibidor. (2) LePro va a seleccionar el lugar donde va a ocurrir el acoplamiento. En otras palabras, selecciona la región de una de las moléculas con “HETATM” en sus respectivas líneas y, luego, borra todas las moléculas con “HETATM” (incluida la que está en ese lugar del acoplamiento). Si no hay “HETATM”, va a seleccionar toda la proteína como lugar de acoplamiento. (3) Para conservar las moléculas que están cerca del lugar de acoplamiento (es decir, los co-factores deseados), hay que cambiar sus respectivos “HETATM” a “ATOM ” (esta última tiene dos espacios luego de la M). (4) Antes de editar el archivo “.pdb”, se tienen que borrar las moléculas (con ChimeraX) no deseadas. En otras palabras, se borran los co-factores que no se van a utilizar y se deja el inhibidor si se quiere que en ese lugar se haga

el análisis.

Por lo tanto, el siguiente paso es utilizar LePro, dependiendo de lo que se quiere se sigue la opción “a” o la “b” que están a continuación. Para ambos casos, los comandos se corren en una ubicación donde estén 6aje_A.pdb y MD_6aje. Se siguieron ambas opciones para comparar como LeDock trabaja con y sin los co-factores (ver p.58, luego de “pro.pdb”).

a) Si no se quiere tener en cuenta a los co-factores, se borran en ChimeraX (estos son “ORO” y “FMN”) antes de generar a 6aje_A.pdb. Y se corre lo siguiente.

```
$ mkdir Docking_6aje;lepro 6aje_A.pdb;cp pro.pdb MD_6aje/MD.pdb;mv pro.pdb dock.in  
Docking_6aje
```

b) Si se quiere tener en cuenta a los co-factores, se corre lo siguiente. El primer comando va a editar al archivo 6aje_A.pdb para que LePro los conserve: para utilizarlo en otro complejo, se tienen que cambiar el nombre de los co-factores (estos son “ORO” y “FMN”) del comando por los correspondientes y se tiene que cambiar las dos veces que aparece 6aje_A.pdb por el archivo correspondiente.

```
$ sed ""$(for w in ORO FMN;do grep -n $w 6aje_A.pdb|grep 'HETATM'|cut -d : -f1;done|  
sed ':a;N;$!ba;s/\n/ /g')"ba;b;a;s/HETATM/ATOM /g" 6aje_A.pdb>6aje_cof.pdb  
$ mkdir Docking_6aje;lepro 6aje_A.pdb;mv pro.pdb MD_6aje/MD.pdb;lepro  
6aje_cof.pdb;mv pro.pdb dock.in Docking_6aje
```

Las opciones “a” y “b” crearon, cada una, los tres archivos pro.pdb, dock.in y MD.pdb. El último está en la carpeta MD_6aje, y contiene la información de la proteína sin co-factores y sin inhibidor. Los dos primeros están en Docking_6aje y contienen información para el programa LeDock. El primero tiene la estructura de la proteína con los co-factores seleccionados. Mientras que el segundo contiene parámetros, incluyendo las coordenadas del sitio activo, es decir, del lugar donde va a ocurrir el acoplamiento.

Lo que continua es preparar los medicamentos para LeDock. Se colocan los archivos “.sdf” descargados en la nueva carpeta 3D_sdf. En esta, se corre lo siguiente. En

https://openbabel.org/docs/dev/Command-line_tools/babel.html hay información respecto a los comandos de OpenBabel.

```
$ mkdir 3D_mol2;conda activate gromacs;for f in *.sdf;do obabel "$f" -O 3D_mol2/a"$f".sdf}_".mol2 -m;done
$ cd 3D_mol2;for f in a*.mol2; do mv $f $(sed -n '2p' $f).mol2;done
```

Esto generó una carpeta 3D_mol2 que contiene archivos “.mol2” con su respectivo código CID de PubChem de nombre, por ejemplo, 5351322.mol2. Luego, en 3D_mol2, se corrió lo siguiente.

```
$ for f in *.mol2;do awk '/@\<TRIPOS>ATOM/{f=1;next} /@\<TRIPOS>BOND/{f=0} f $f|awk '{print $6}'|sed 's/\.\*/g'|awk '!seen[$0]++>>a;done;awk '!seen[$0]++ a;rm a
$ echo CID,C,H,N,O,Si,P,S,F,Cl,Br,I>../CID.csv;for f in *.mol2;do awk '/@\<TRIPOS>ATOM/{f=1;next} /@\<TRIPOS>BOND/{f=0} f $f|awk '{print $6}'|sed
's/\.\*/g>a;echo ${f%.mol2},${(grep -c '^C$' a)}${(grep -c '^H$' a)}${(grep -c '^N$' a)}${(grep -c '^O$' a)}${(grep -c '^Si$' a)}${(grep -c '^P$' a)}${(grep -c '^S$' a)}${(grep -c '^F$' a)}${(grep -c '^Cl$' a)}${(grep -c '^Br$' a)}${(grep -c '^I$' a)}>>../CID.csv;rm a;done
```

Ambos comandos extraen información de los archivos “.mol2” y pueden estar sujetos a errores. El primero muestra en la Terminal los elementos presentes en estos archivos y el segundo comando creó, en 3D_sdf, el archivo CID.csv, el cual contiene una lista de los medicamentos con la cantidad respectiva de cada elemento. Si los elementos presentes no están incluidos en C, H, N, O, Si, P, S, F, Cl, Br e I, se tienen que editar el comando.

El archivo CID.csv se puede abrir en una hoja de cálculo para calcular la masa atómica de cada medicamento, lo cual a su vez permite filtrar más a los medicamentos.

Se seleccionaron medicamentos que contengan sólo los átomos C, N, H y O, hasta 5 oxígenos, y que tengan valores mayores a 200 Da (para los buscados directamente por PubChem). Los nombres de estos se guardaron (uno por línea) en un nuevo archivo de nombre s_list.txt dentro de 3D_sdf. Luego, se corrió en esta lo siguiente.

```
$ mkdir s_mol2;for f in $(cat s_list.txt);do cp 3D_mol2/$f.mol2 s_mol2;done
```

Después, los archivos de s_mol2 se copiaron en Docking_6aje. Además, se incluyó al

inhibidor en formato “.mol2”. Pero, como estaba en formato “.pdb”, se corrió, en MD_6aje, el comando “\$ conda activate gromacs;obabel 1_ferulenol.pdb -O 1_ferulenol.mol2”. Después, se movió el archivo “.mol2” creado a la carpeta Docking_6aje. Otra opción es buscar el inhibidor en internet y descargarlo, lo cual también se realizó.

Luego, se utiliza LeDock. Sin embargo, este programa usa sólo un hilo del CPU, por lo tanto se corrió lo siguiente en Docking_6aje, donde NUMERO es igual al número de hilos a utilizar y APAGADO es un comando opcional de alarma o de apagado. Como “i3” tiene cuatro hilos, se utilizó el número 3, lo cual permite tener un hilo para otras funciones.

```
$ n=NUMERO;a=$((($((ls -l *mol2|wc -l)+$n-1))/$n);i=0; for f in *.mol2;do d=dir_$
  (printf %03d $((i/$a+1)));mkdir -p $d;mv "$f" $d; let i++;done
$ for d in dir_*;do cp dock.in pro.pdb $d;done
$ for d in dir_*;do cd $d;echo $d,$(date)>>../time_ledock.txt&&ls *.mol2>ligands&&ledock
  dock.in|tee -a ../time_ledock.txt&&echo
  $d,$(date)>>../time_ledock.txt&&cd ..;done;wait;APAGADO
$ mv dir_*/*.{mol2,dok} .
$ rm -r dir_*
$ echo CID,kcal/mol>LeDock.csv;for f in *.dok;do a=$(echo ${f%.dok});b=$(sed -n '2p' $f
  sed 's/^.*Score: //;s/ kcal/mol//');echo $a,$b>>LeDock.csv;done
```

El tercer comando creó un archivo “.dok” por cada medicamento y el archivo time_ledock.txt. Este último tiene información del inicio y el final de cada proceso, y de la Terminal (si hay co-factores pueden aparecer mensajes de “Warning: Missing Parameters for Residue”). Cabe mencionar que se realizaron pruebas con y sin co-factores en el archivo pro.pdb: cuando hay co-factores los valores de energía son más negativos, pero se mantiene un orden similar y las posiciones de acoplamiento son prácticamente equivalentes.

Por otro lado, el último comando creó a LeDock.csv. Este tiene los mejores puntajes de los medicamentos extraídos de los “.dok” y en valores de energía de acoplamiento (kcal/mol): mientras menor sea la energía, mejor es el acoplamiento. Este archivo se puede abrir en una hoja de cálculo para seleccionar a las mejores opciones. Para seleccionarlas, se pueden ordenar mediante filtros la energía de acoplamiento de manera ascendente (es decir, arriba

van a estar las más negativas) y, también, se pueden usar las masas atómicas obtenidas mediante CID.csv. Cabe mencionar que LeDock tiene un sesgo a favor de moléculas más pesadas o con ciertos grupos funcionales.

Conviene seleccionar algunos medicamentos (por ejemplo, 300) para generar duplicados (evita volver a correr todos varias veces). Para este fin, dentro de Docking_6aje, se creó la carpeta m_ledock y, en esta última, a un archivo m_ledock.txt que contiene los nombres de los medicamentos seleccionados (uno por línea). Luego, en m_ledock, se corrió lo siguiente.

```
$ mkdir m_ledock_01;cp $(for f in $(cat m_ledock.txt);do echo .."$f".dok;
echo .."$f".mol2;done) ../{dock.in,pro.pdb} m_ledock_01;cd m_ledock_01;ls
*.mol2>ligands;cd ..
$ for ((i=0;i<DUPLICADOS;i++)); do mkdir m_ledock_$(printf %02d $((i+2)));cp
m_ledock_01/{*.mol2,dock.in,pro.pdb,ligands} m_ledock_$(printf %02d $((i+2)));done
$ for d in m_ledock_[0-9][0-9];do if [ $d == m_ledock_01 ]; then continue;fi;cd $d;echo $d,$
(date)>>../time_ledock.txt&&ls *.mol2>ligands&&ledock dock.in|tee -a
../time_ledock.txt&&echo $d,$(date)>>../time_ledock.txt&&cd ../done;wait;APAGADO
$ echo CID,01>LeDock_d.csv;cd m_ledock_01;for f in *.mol2;do a=$(echo ${f
%.mol2});b=$(sed -n '2p' ${f%.mol2}.dok|sed 's/^.*Score: //;s/ kcal/mol//');echo $a,
$b>>../LeDock_d.csv;done;cd ..
$ for d in m_ledock_[0-9][0-9];do if [ $d == m_ledock_01 ]; then continue;fi;cd $d;echo ,${
d#m_ledock_}>a.csv;for f in *.dok;do echo ,$(sed -n '2p' $f|sed 's/^.*Score: //;s/
kcal/mol//')>>a.csv;done;paste -d "" ../LeDock_d.csv a.csv>b.csv;cat
b.csv>../LeDock_d.csv;rm [ab].csv;cd ../done
```

En el segundo comando, DUPLICADOS corresponde al número de corridas adicionales. Por ejemplo, se utilizó el número 3, con lo cual se van a obtener cuatro valores (es decir, el original más los tres nuevos). Estos valores van a estar en LeDock_d.csv, el cual se puede abrir en una hoja de cálculo para calcular su promedio, ordenarlos y añadir información de otros “.csv”. Cabe mencionar que el tercer comando es para el caso de que el número de duplicados sea igual al número de hilos a utilizar, de lo contrario es necesario modificarlo para optimizar su uso (además, genera otro archivo time_ledock.txt).

Luego, se seleccionaron medicamentos teniendo en cuenta a LeDock_d.csv y a información complementaria: se copiaron sus archivos “.dok” (al menos uno de los generados si se

realizaron duplicados) en una nueva carpeta m_pdb. En esta se corrió lo siguiente para producir, por cada medicamento, una carpeta con su nombre, la cual contiene los archivos “.pdb” de sus posiciones de acoplamiento, enumeradas de mejor a peor valor.

```
$ for f in *.dok;do ledock -spli $f;mkdir ${f%.dok};mv *pdb ${f%.dok};done
```

Finalmente, se seleccionaron los archivos “.pdb” de los medicamentos que van a ser analizados en DM y se copian a MD_6aje. Para visualizar las posiciones y seleccionarlas se puede utilizar ChimeraX: se puede optar por una posición con peor valor de un mismo medicamento. La información dentro de MD_6aje va a ser utilizada en el siguiente paso. Si se quiere utilizar l_ferulenol.pdb y no una de sus posiciones generadas por LeDock, se tiene que correr “\$ sed -i 's/9AU/LIG/g' l_ferulenol.pdb”.

A.2.3. Trayectorias de DM

Luego de abrir en ChimeraX el archivo [L]_md.gro con su trayectoria, en ese orden, se puede guardar un video si se corre, en la barra de comandos de ChimeraX, primero, “movie record limit 100” (para 1 ns) y, luego, “movie encode” (genera un video mp4). Se puede usar la Terminal para correr el comando “\$ ffmpeg -i IN.mp4 -an -vf "setpts=2.0*PTS" OUT.mp4”, lo cual reduce la velocidad de reproducción a la mitad (cambiar “IN” y “OUT”), puede que se requieran instalaciones para correr este comando.

Sin embargo, la gráfica de rmsd.csv puede mostrar un periodo de tiempo de interés. Por lo tanto, ya que ChimeraX no puede abrir [L]_s.xtc (muy larga al ser de 100 ns), se tiene que extraer el periodo de tiempo deseado con el siguiente comando en c_f. En el comando se tiene que remplazar los [T_INICIAL] y [T_FINAL] por valores en picosegundos y los [L] por el nombre del medicamento deseado. El archivo resultante va a aparecer en la carpeta R con el nombre [L]_b[T_INICIAL]e[T_FINAL].xtc.

```
$ source ~/gromacs/bin/GMXRC;echo System|gmx trjconv -b [T_INICIAL] -e [T_FINAL] -
n [L]_em.ndx -f mm/[L]_s.xtc -s md/[L]_md.tpr -o R/[L]_b[T_INICIAL]e[T_FINAL].xtc
```

Finalmente, si se quiere ampliar el tiempo de al menos una simulación, se crea, en la carpeta c_f, el archivo de texto plano ampliar.txt que contiene los nombres (uno por línea) de los medicamentos con simulación a ampliar y se corre el siguiente comando.

```
$ mkdir -p Tampliado/{md,mm,R};for f in $(cat ampliar.txt);do mv md/"$f"*
R/log/"$f"_md.log Tampliado/md;mv "$f"_*.[itn][tod][px] Tampliado;done;cp mm/mm.in
Tampliado/mm;cp cof_*.itp posre.itp t.txt Tampliado
```

Luego, se continúa con la instrucciones de 2b.sh (ver “A.3.3.”).

A.3. Archivos “.sh”

Cada subcapítulo de esta parte muestra los contenidos de a.sh, b.sh, c.sh y 2b.sh, respectivamente. En los comentarios de cada archivo están las instrucciones a seguir y lo que hace cada parte del código.

A.3.1. a.sh

```
# En una carpeta vacía realizar los pasos de las siguientes cuatro líneas.
# Guardar en ella este archivo “a.sh”
# Guardar una copia del archivo “MD.pdb” (de la proteína a trabajar)
# Crear una carpeta de nombre “L”, y guardar en esta cada medicamento y, si hay, los
  co-factores de la proteína. Todas estas moléculas tienen que estar en formato “.pdb” y sólo
  cada co-factor debe tener “cof_” antes de su nombre.
# Ejecutar “a.sh” con la siguiente línea.

# (bash -i a.sh) 2>&1|tee -a t.txt

# En “t.txt” se van a guardar los comentarios de la ejecución.
# Una alternativa es agregar un comando de apagado y una alarma, como en el siguiente
  ejemplo.
# (bash -i a.sh) 2>&1|tee -a t.txt;shutdown +5;vlc ~/Desktop/00_Programas/beeping.mp3
# “shutdown -c” cancela este comando de apagado.

# De ser necesario editar en “# Se crean parámetros”.
# A continuación está lo que “a.sh” va a ejecutar y su descripción.

# Se activa el ambiente.
```

```

date
set -e
conda activate gromacs;source ~/gromacs/bin/GMXRC

# Se crea la topología de la proteína. Primero, se edita el archivo “.pdb” de la proteína creado
# por LePro para que pueda ser leído por GROMACS. Luego, se utiliza “gmx pdb2gmx” para
# crear tres archivos: conf.gro, posre.itp y topol.top cpx.
sed -i 's/HSD/HIS/g;s/HSE/HIS/g' MD.pdb
gmx pdb2gmx -ff amber99sb-ildn -water tip3p -ignh -f MD.pdb
mkdir cpx;mv conf.gro posre.itp topol.top cpx

# Se crea la topología de los co-factores (si hay) y de cada medicamento. Primero, se utiliza
# ACPYPE para crear los archivos “[L]_GMX.gro” y “[L]_GMX.itp” de cada molécula
# (“[L]” representa el nombre de la molécula). Luego, “gmx genrestr” crea los archivos
# “[L]_posre.itp” usados para restringir el movimiento de su respectiva molécula. Finalmente,
# “[L]_GMX.itp” es dividido en “[L]_1.itp” y “[L]_2.itp”.
cd L
for f in *.pdb
do acpype -i "$f";mv "${f%.pdb}.acpype/${f%.pdb}_GMX."{gro,itp} ../cpx
done
cd ../cpx
for f in *_GMX.gro
do (echo ""System" & ! a H*;echo q)|gmx make_ndx -f "$f" -o "${f%_GMX.gro}.ndx"
echo 'System_&!H*'gmx genrestr -n "${f%_GMX.gro}.ndx" -f "$f" -o "${f
%GMX.gro}posre.itp"
a=$(grep -n '\[ moleculetype \]' "${f%gro}itp"|cut -d : -f1)
sed -n "3,${a-2}p" "${f%gro}itp">"${f%GMX.gro}1.itp"
sed -n "${a,$ p" "${f%gro}itp">"${f%GMX.gro}2.itp"
done

# Se crea la topología de los complejos: proteína + co-factores (si hay) + medicamento
# Primero, se remueve la última línea de “conf.gro” y se guarda en “conf.txt”.
# Luego, si hay co-factores, su información se agrega a “topol.top”, a “conf.gro” (de
# “[L]_GMX.gro”) y a “b.txt” (de “[L]_1.itp”). La información del “b.txt” se va a agregar a
# cada “[L]_1.itp” de los medicamentos.
# Finalmente, para los medicamentos, se crea “[L]_cpx.top” (con “topol.top” e información
# del respectivo [L]) y “[L]_cpx.gro” (con “conf.gro”, “[L]_GMX.gro” y “conf.txt”).
tail -1 conf.gro>conf.txt;sed -i '$d' conf.gro
for f in cof_*GMX.gro; do test -f $f && >b.txt && mkdir cof && mv cof_*GMX.[gin][rtd]
[opx] cof && break; done
if test -d cof
then for f in cof_*1.itp
do cat topol.top|sed "$ (grep -n '; Include water topology' topol.top|cut -d : -f1)i; Include
cofactor topology\n#include \"\"${f%1.itp}2.itp\"\"\n\n#ifdef POSRES\n#include \"\"$ {f
%1.itp}posre.itp\"\"\n#endif\n\">a.txt
echo \"\"${f%_1.itp} \" 1\">>a.txt
cat a.txt>topol.top
a=$(sed -n '2p' cof/"${f%1.itp}GMX.gro")

```

```

b=$(sed -n '2p' conf.gro)
cat conf.gro|sed "2s/.*/$(($a+$b))/">a.txt
cat cof/"${f%1.itp}GMX.gro"|sed '1,2d'|sed '$d'>>a.txt
cat a.txt>conf.gro
cat $f>>b.txt;mv $f cof
done
fi
for f in *_GMX.gro
do cat topol.top|sed "$ (grep -n ' Include water topology' topol.top|cut -d : -f1)i; Include
ligand topology\n#include \"\"${f%GMX.gro}2.itp\"\"\n\n#ifndef POSRES\n#include \"\"${f
%GMX.gro}posre.itp\"\"\n#endif\n"|sed "/forcefield\\.itp"/a#include \"\"${f
%GMX.gro}1.itp\"\">"${f%GMX.gro}cpx.top"
echo ""${f%_GMX.gro}" 1">>"${f%GMX.gro}cpx.top"
a=$(sed -n '2p' "$f")
b=$(sed -n '2p' conf.gro)
cat conf.gro|sed "2s/.*/$(($a+$b))/">"${f%GMX.gro}cpx.gro"
cat "$f"|sed '1,2d'|sed '$d'>>"${f%GMX.gro}cpx.gro"
cat conf.txt>>"${f%GMX.gro}cpx.gro"
if test -d cof
then cat b.txt>>"${f%GMX.gro}1.itp";awk '!x[$0]++' "${f%GMX.gro}1.itp">a.txt;cat
a.txt>"${f%GMX.gro}1.itp"
fi
done
mkdir ../I;mv !(I*_cpx.???|*_ [12].itp)*posre.itp) ../I

# Se crean parámetros, dentro de 6 archivos: ions.mdp, em.mdp, nvt.mdp, npt.mdp, md.mdp
y mm.in
# Si hay co-factores, a cada "Protein_LIG" se le agregan los nombres de los co-factores
separados por "_", por ejemplo en "tc-grps = Protein_LIG_FMN_ORO Water_and_ions".
# "nsteps = 500000" es 1ns de simulación. En md.mdp, se usa 50000000, que es 100ns.

cat<<EOF>ions.mdp
integrator = steep
emtol = 1000.0
emstep = 0.01
nsteps = 50000
cutoff-scheme = Verlet
ns-type = grid
nstlist = 1
rlist = 1.0
coulombtype = cutoff
rcoulomb = 1.0
rvdw = 1.0
pbc = xyz
EOF

cat<<EOF>em.mdp
integrator = steep

```

```
emtol = 1000.0
emstep = 0.01
nsteps = 50000
cutoff-scheme = Verlet
ns-type = grid
nstlist = 1
rlist = 1.0
vdwtype = cutoff
vdw-modifier = Potential-shift-Verlet
rvdw-switch = 1.0
rvdw = 1.0
coulombtype = P3M-AD
rcoulomb = 1.0
pbc = xyz
DispCorr = EnerPres
EOF
```

```
cat<<EOF>nvt.mdp
define = -DPOSRES
integrator = md
nsteps = 50000
dt = 0.002
nstenergy = 500
nstlog = 500
nstxout-compressed = 500
continuation = no
constraint-algorithm = lincs
constraints = h-bonds
lincs-iter = 1
lincs-order = 4
cutoff-scheme = Verlet
ns-type = grid
nstlist = 20
rlist = 1.0
vdwtype = cutoff
vdw-modifier = Potential-shift-Verlet
rvdw-switch = 1.0
rvdw = 1.0
coulombtype = P3M-AD
rcoulomb = 1.0
pme-order = 4
fourierspacing = 0.125
tcoupl = v-rescale
tc-grps = Protein_LIG Water_and_ions
tau-t = 0.1 0.1
ref-t = 300 300
pcoupl = no
pbc = xyz
```

```
DispCorr = EnerPres
gen-vel = yes
gen-temp = 300
gen-seed = -1
EOF
```

```
cat<<EOF>npt.mdp
define = -DPOSRES
integrator = md
nsteps = 50000
dt = 0.002
nstenergy = 500
nstlog = 500
nstxout-compressed = 500
continuation = yes
constraint-algorithm = lincs
constraints = h-bonds
lincs-iter = 1
lincs-order = 4
cutoff-scheme = Verlet
ns-type = grid
nstlist = 20
rlist = 1.0
vdwtype = cutoff
vdw-modifier = Potential-shift-Verlet
rvdw-switch = 1.0
rvdw = 1.0
coulombtype = P3M-AD
rcoulomb = 1.0
pme-order = 4
fourierspacing = 0.125
tcoupl = v-rescale
tc-grps = Protein_LIG Water_and_ions
tau-t = 0.1 0.1
ref-t = 300 300
pcoupl = Berendsen
pcoupltype = isotropic
tau-p = 2.0
ref-p = 1.0
compressibility = 4.5e-5
refcoord-scaling = com
pbc = xyz
DispCorr = EnerPres
gen-vel = no
EOF
```

```
cat<<EOF>md.mdp
integrator = md
```

```

nsteps = 50000000
dt = 0.002
nstenergy = 5000
nstlog = 5000
nstxout-compressed = 5000
continuation = yes
constraint-algorithm = lincs
constraints = h-bonds
lincs-iter = 1
lincs-order = 4
cutoff-scheme = Verlet
ns-type = grid
nstlist = 20
rlist = 1.0
vdwtype = cutoff
vdw-modifier = Potential-shift-Verlet
rvdw-switch = 1.0
rvdw = 1.0
coulombtype = P3M-AD
rcoulomb = 1.0
pme-order = 4
fourierspacing = 0.125
tcoupl = v-rescale
tc-grps = Protein_LIG Water_and_ions
tau-t = 0.1 0.1
ref-t = 300 300
pcoupl = Parrinello-Rahman
pcoupltype = isotropic
tau-p = 2.0
ref-p = 1.0
compressibility = 4.5e-5
pbc = xyz
DispCorr = EnerPres
gen-vel = no
EOF

cat<<EOF>mm.in
&general
sys_name="Prot-Lig-ST",
startframe=5,
endframe=14,
forcefields="oldff/leaprc.ff99SB,leaprc.gaff"
/
&gb
igb=5, saltcon=0.150,
/
EOF

```

En la carpeta donde se ejecutó a.sh van a haber 3 carpetas (cpx, I y L) y 3 archivos (t.txt, MD.pdb y a.sh). Sólo cpx y t.txt van a ser utilizados en los siguientes pasos. “cpx” va a contener 7 archivos en general (posre.itp, ions.mdp, em.mdp, nvt.mdp, npt.mdp, md.mdp y mm.in), 2 archivos por co-factor ([L]_2.itp y [L]_posre.itp), 5 archivos por medicamento ([L]_1.itp, [L]_2.itp, [L]_cpx.gro, [L]_cpx.top y [L]_posre.itp). Por ejemplo, si se analizan 6 medicamentos para una proteína con 2 co-factores, “cpx” va a contener 41 archivos.

date

A.3.2. b.sh

“cpx”, “t.txt” y este archivo (i.e., “b.sh”) se colocan en una nueva carpeta. En esta se corre el siguiente comando.

```
# (bash -i b.sh) 2>&1|tee -a t.txt;mv t.txt c_f
```

Si no se ejecuta bien b.sh, puede que t.txt cambie de nombre a c_f en lugar de moverse a una carpeta con ese nombre.

De ser necesario, editar en “# Se minimiza la energía”.

A continuación está lo que “b.sh” va a ejecutar y su descripción.

date

set -e

source ~/gromacs/bin/GMXRC

cd cpx

Se genera solución alrededor del complejo. Mejor dicho, una caja de simulación en forma de dodecaedro rómbico con condiciones de frontera periódicas que contiene al complejo rodeado por una solución de NaCl 0.15 mol/L.

GROMACS va a mostrar mensajes de advertencia en este paso y en los siguientes, hasta antes de simulación de 100ns, ya que se están preparando las condiciones para esta última.

Los [L]_cpx.gro, producen a los [L]_cpx4em.gro.

```
for f in *cpx.top
```

```
do gmx editconf -bt dodecahedron -d 1.0 -f "${f%.top}.gro" -o "${f%.top}.b.gro"
```

```
gmx solvate -cs spc216.gro -cp "${f%.top}.b.gro" -p "$f" -o "${f%.top}.4ion.gro"
```

```
gmx grompp -f ions.mdp -c "${f%.top}.4ion.gro" -p "$f" -o "${f%.top}.4ion.tpr"
```

```
echo SOL|gmx genion -conc 0.15 -neutral -s "${f%.top}.4ion.tpr" -p "$f" -o "${f
```

```
%top}.4em.gro"
```

```
done
```

```
mkdir sol;mv \##\# *[xbn].[gt][rp][or] ions.mdp mdout.mdp sol
```

Se minimiza la energía.

Hay un comentario importante entre los comandos de esta parte.

Se crean archivos [L]_em.gro, [L]_em.ndx (un índice) y em_PE.csv. El último está en una nueva carpeta R y tiene los resultados, generados por la minimización, de cada energía potencial del complejo.

```
mkdir em R;for f in *cpx.top
```

```
do gmx grompp -f em.mdp -c "${f%.top}.4em.gro" -p "$f" -o em/"${f%.cpx.top}.em.tpr" -
```

```
maxwarn 1
```

```

done
cd em;for f in *.tpr
do gmx mdrun -deffnm "${f%.tpr}"
(echo "'Protein'|'LIG'";echo q)|gmx make_ndx -f "${f%.tpr}.gro" -o ../"${f%.tpr}.ndx"
# Si se tiene co-factores, agregar a "Protein"| "LIG" los nombres de los co-factores separados
por "|", por ejemplo "(echo "'Protein'|'LIG'|'FMN'|'ORO'";". Lo que viene después de
este "echo", está entre comillas simples.
done
for f in *_em.log
do a=$(grep 'Potential Energy = ' "$f"|cut -c 21-)
b=$(grep 'Steepest Descents converged to Fmax < 1000 in ' "$f"|cut -c 47-|sed 's/ steps//')
echo "${f%.em.log},${a},${b}">>../R/em_PE.csv
done;cd ../;sed -i '1i\em.log,PE(kJ/mol),Converged_Fmax<1000kJ/mol_(steps)'
R/em_PE.csv;mv *em.[mg][dr][po] mdout.mdp em

# Se simuló el sistema a volumen constante y, luego, a presión constante. En ambos casos, el
número de partículas y la temperatura también fueron constantes. Por eso se utilizan las
nombres nvt y npt, respectivamente.
set +e
mkdir nvt;for f in *cpx.top
do gmx grompp -n "${f%.cpx.top}.em.ndx" -f nvt.mdp -c em/"${f%.cpx.top}.em.gro" -r em/"$
{f%.cpx.top}.em.gro" -p "$f" -o nvt/"${f%.cpx.top}.nvt.tpr"
cd nvt
date
gmx mdrun -deffnm "${f%.cpx.top}.nvt";if [ ! -e *gro ];then gmx mdrun -deffnm "${f
%cpx.top}.nvt" -cpi "${f%.cpx.top}.nvt_prev.cpt";fi
date
cd ../;done;mv nvt.mdp mdout.mdp nvt

mkdir npt;for f in *cpx.top
do gmx grompp -t nvt/"${f%.cpx.top}.nvt.cpt" -n "${f%.cpx.top}.em.ndx" -f npt.mdp -c nvt/"$
{f%.cpx.top}.nvt.gro" -r nvt/"${f%.cpx.top}.nvt.gro" -p "$f" -o npt/"${f%.cpx.top}.npt.tpr" -
maxwarn 1
cd npt
date
gmx mdrun -deffnm "${f%.cpx.top}.npt";if [ ! -e *gro ];then gmx mdrun -deffnm "${f
%cpx.top}.npt" -cpi "${f%.cpx.top}.npt_prev.cpt";fi
date
cd ../;done;mv npt.mdp mdout.mdp npt

# Se realizó la corrida de producción.
# Cabe mencionar que md.mdp no utiliza restricciones de posiciones (i.e., no tiene "define = -
DPOSRES")
mkdir md;for f in *cpx.top
do gmx grompp -t npt/"${f%.cpx.top}.npt.cpt" -n "${f%.cpx.top}.em.ndx" -f md.mdp -c npt/"$
{f%.cpx.top}.npt.gro" -p "$f" -o md/"${f%.cpx.top}.md.tpr"
cd md
date

```

```

gmx mdrun -deffnm "${f%cp*.top}.md";if [ ! -e *gro ];then gmx mdrun -deffnm "${f
%cp*.top}.md" -cpi "${f%cp*.top}.md_prev.cpt";fi
date
mv "${f%cp*.top}.md.gro" ../R
cd ..;done;mv md.mdp mdout.mdp md
set -e

# Se procesó la trayectoria de la corrida de producción.
# La trayectoria original, [L]_md.xtc, se modifica a [L]_s.xtc para que pueda ser visualizada.
# Luego, [L]_s.xtc genera los archivos [L]_i1s.xtc y [L]_f1s.xtc . Los cuales son el primero y
# el último 1 ns, respectivamente.
mkdir mm;mv mm.in mm;for f in *em.ndx
do (echo Protein;echo System)|gmx trjconv -pbc mol -ur compact -center -n "$f" -f md/"${f
%em.ndx}.md.xtc" -s md/"${f%em.ndx}.md.tpr" -o mm/"${f%em.ndx}4s.xtc"
(echo Backbone;echo System)|gmx trjconv -fit rot+trans -n "$f" -f mm/"${f%em.ndx}4s.xtc"
-s md/"${f%em.ndx}.md.tpr" -o mm/"${f%em.ndx}s.xtc"
echo System|gmx trjconv -b 99000 -n "$f" -f mm/"${f%em.ndx}s.xtc" -s md/"${f
%em.ndx}.md.tpr" -o R/"${f%em.ndx}f1s.xtc"
echo System|gmx trjconv -e 1000 -n "$f" -f mm/"${f%em.ndx}s.xtc" -s md/"${f
%em.ndx}.md.tpr" -o R/"${f%em.ndx}i1s.xtc"
done

# RMSD y RMSF
# Se crean los archivos rmsd.csv y rmsf.csv en la carpeta R. El primero se usa para analizar la
# estabilidad del complejo. Mientras que la información del segundo sirve para comparar lo
# conformación de los complejos de los medicamento con la del complejo del inhibidor.
mkdir rmsd;for f in *em.ndx
do (echo Backbone;echo LIG)|gmx rms -tu ns -what rmsd -n "$f" -f mm/"${f%em.ndx}s.xtc"
-s md/"${f%em.ndx}.md.tpr" -o rmsd/"${f%em.ndx}rmsd.xvg"
done
cd rmsd;a=$(sed -n '19p' $(ls *xvg|head -1));b="${a%.*}";c=$(( ${#b} - 1 ));cat $(ls *xvg|head
-1)|sed '1,18d'|cut -c -$c|sed 's/ //g'>../R/rmsd.csv
for f in *xvg
do cat "$f"|sed '1,18d'|cut -c $c->a.txt;paste -d ' ' ../R/rmsd.csv a.txt >b.txt;cat
b.txt>../R/rmsd.csv
done
rm a.txt b.txt;sed -i "s/\s\+//g;i\`ns,\"$(ls -m *xvg|tr -d ' \n')\"" ../R/rmsd.csv;cd ..

mkdir rmsf;for f in *em.ndx
do echo C-alpha|gmx rmsf -res -n "$f" -f R/"${f%em.ndx}f1s.xtc" -s md/"${f
%em.ndx}.md.tpr" -o rmsf/"${f%em.ndx}rmsf.xvg"
done
cd rmsf;a=$(sed -n '18p' $(ls *xvg|head -1));b="${a%.*}";c=$(( ${#b} - 1 ));cat $(ls *xvg|head
-1)|sed '1,17d'|cut -c -$c|sed 's/ //'>../R/rmsf.csv
for f in *xvg
do cat "$f"|sed '1,17d'|cut -c $c->a.txt;paste -d ' ' ../R/rmsf.csv a.txt >b.txt;cat
b.txt>../R/rmsf.csv
done

```



```
cd cpx
```

```
# Se amplía la simulación
```

```
# Cambiar TIEMPO_A al tiempo a añadir en pico-segundos (e.g., 50000).
```

```
cd md;for f in *_md.tpr
```

```
do date;gmx convert-tpr -s $f -extend TIEMPO_A -o $f;gmx mdrun -deffnm ${f%.tpr} -cpi $  
  ${f%.tpr}.cpt;if [ ! -e *gro ];then gmx mdrun -deffnm ${f%.tpr} -cpi ${f%.tpr}_prev.cpt;fi  
date;done;cd ..
```

```
set -e
```

```
# Se procesó la trayectoria ampliada de la corrida de producción.
```

```
# El nuevo [L]_md.xtc se modifica a [L]_s.xtc para que pueda ser visualizada. Luego,
```

```
[L]_s.xtc genera el archivo [L]_fls.xtc.
```

```
for f in *em.ndx
```

```
do (echo Protein;echo System)|gmx trjconv -pbc mol -ur compact -center -n "$f" -f md/"${f  
  %em.ndx}md.xtc" -s md/"${f%em.ndx}md.tpr" -o mm/"${f%em.ndx}4s.xtc"
```

```
(echo Backbone;echo System)|gmx trjconv -fit rot+trans -n "$f" -f mm/"${f%em.ndx}4s.xtc"  
  -s md/"${f%em.ndx}md.tpr" -o mm/"${f%em.ndx}s.xtc"
```

```
echo System|gmx trjconv -b TIEMPO_F -n "$f" -f mm/"${f%em.ndx}s.xtc" -s md/"${f  
  %em.ndx}md.tpr" -o R/"${f%em.ndx}fls.xtc"
```

```
# Cambiar TIEMPO_F para que la nueva trayectoria sólo incluya el último ns: sumar a 99000  
el TIEMPO_A (e.g., 149000).
```

```
done
```

```
# RMSD y RMSF
```

```
# Se crean los archivos rmsd.csv y rmsf.csv en la carpeta R.
```

```
mkdir rmsd;for f in *em.ndx
```

```
do (echo Backbone;echo LIG)|gmx rms -tu ns -what rmsd -n "$f" -f mm/"${f%em.ndx}s.xtc"  
  -s md/"${f%em.ndx}md.tpr" -o rmsd/"${f%em.ndx}rmsd.xvg"
```

```
done
```

```
cd rmsd;a=$(sed -n '19p' $(ls *xvg|head -1));b="${a%.*}";c=$(( ${#b} - 1 ));cat $(ls *xvg|head  
  -1)|sed '1,18d'|cut -c -$c|sed 's/ //g'>../R/rmsd.csv
```

```
for f in *xvg
```

```
do cat "$f"|sed '1,18d'|cut -c $c->a.txt;paste -d ' ' ../R/rmsd.csv a.txt >b.txt;cat  
  b.txt>../R/rmsd.csv
```

```
done
```

```
rm a.txt b.txt;sed -i "s/^s\+//g;1i\ns,\"$(ls -m *xvg|tr -d ' \n')\"" ../R/rmsd.csv;cd ..
```

```
mkdir rmsf;for f in *em.ndx
```

```
do echo C-alpha|gmx rmsf -res -n "$f" -f R/"${f%em.ndx}fls.xtc" -s md/"${f  
  %em.ndx}md.tpr" -o rmsf/"${f%em.ndx}rmsf.xvg"
```

```
done
```

```
cd rmsf;a=$(sed -n '18p' $(ls *xvg|head -1));b="${a%.*}";c=$(( ${#b} - 1 ));cat $(ls *xvg|head  
  -1)|sed '1,17d'|cut -c -$c|sed 's/ //'>../R/rmsf.csv
```

```
for f in *xvg
```

```
do cat "$f"|sed '1,17d'|cut -c $c->a.txt;paste -d ' ' ../R/rmsf.csv a.txt >b.txt;cat  
  b.txt>../R/rmsf.csv
```

```
done
```

```
rm a.txt b.txt;sed -i "s/\s\+//g;1i\Re,"$(ls -m *.xvg|tr -d '\n')"' ..R/rmsf.csv;cd ..
```

Se ordenan los archivos.

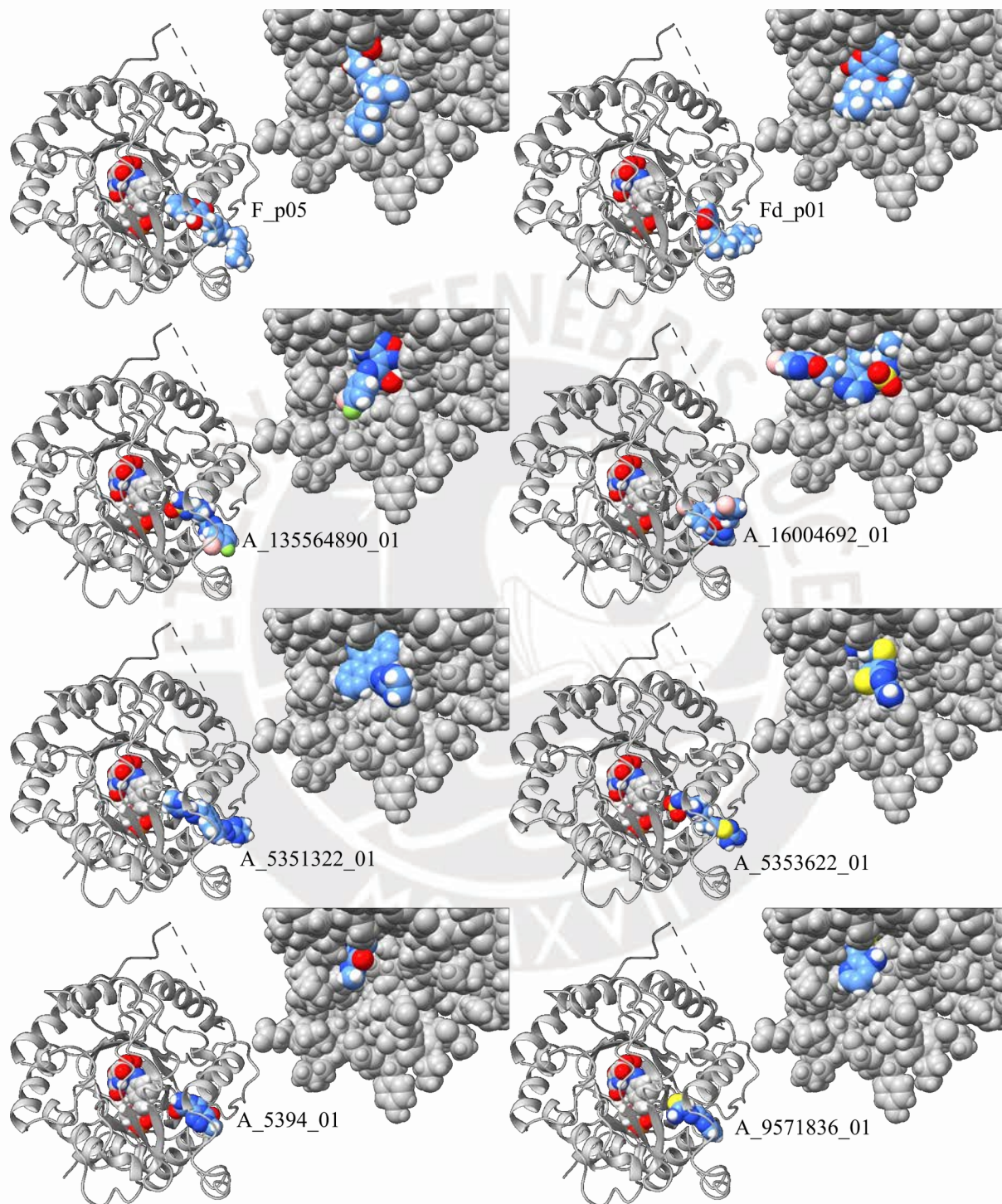
```
mkdir R/log R/csv ../c_f;mv md/*log R/log;mv R/*.csv R/csv  
date
```

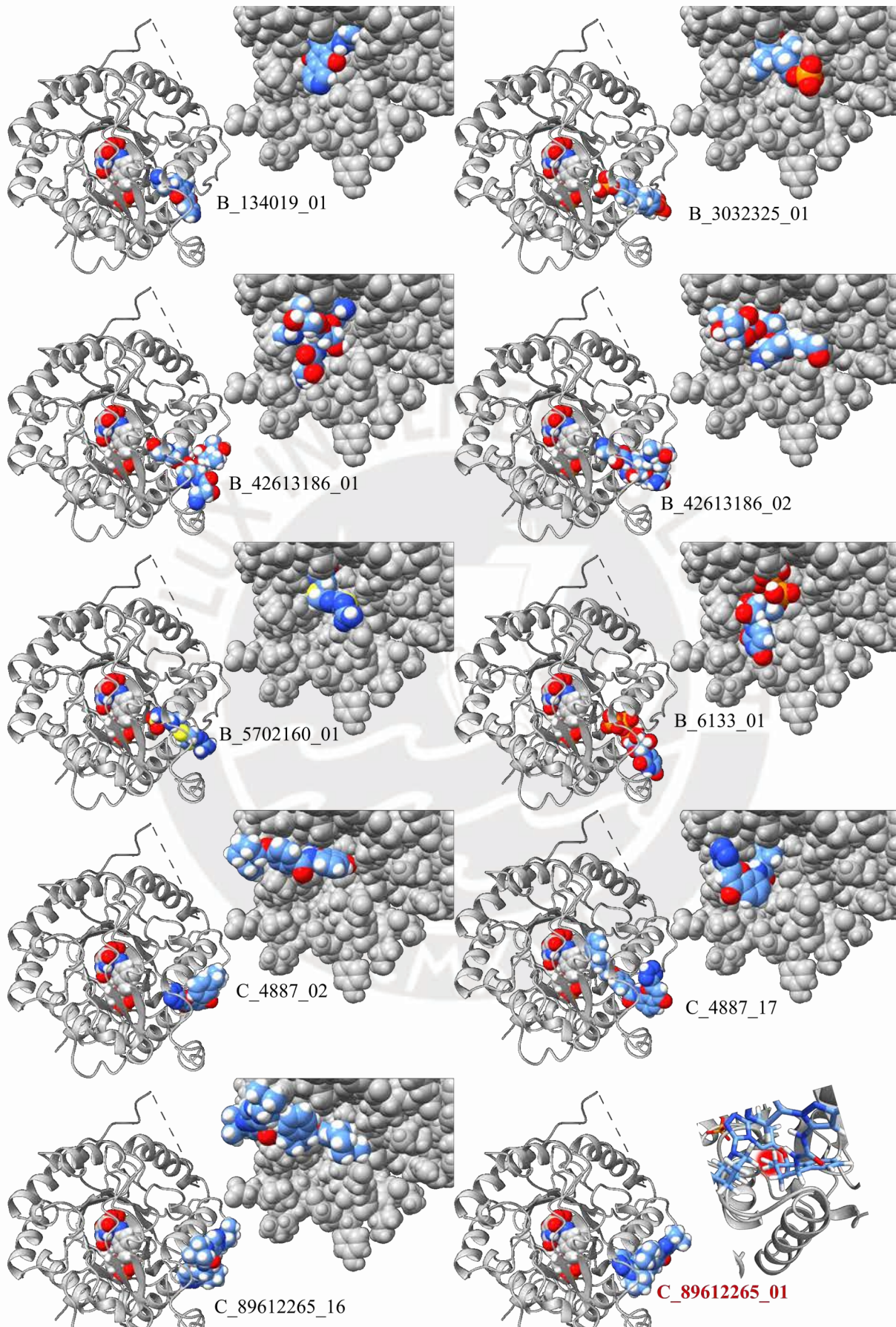
Lo que continua es correr c.sh en esta carpeta (i.e., Tampliado) en lugar de la carpeta c_f.



A.4. Resultados completos

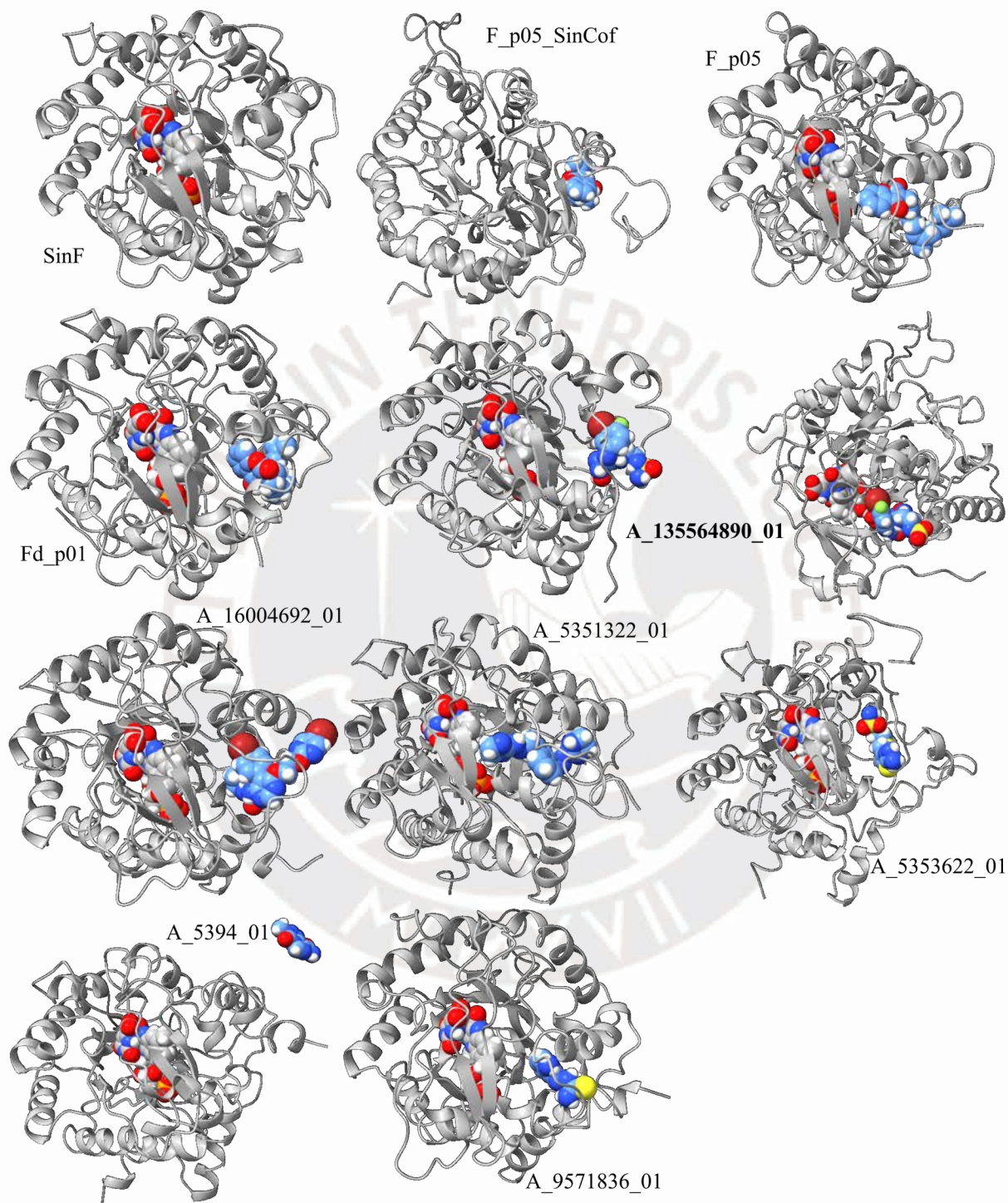
A.4.1. Posiciones de los medicamentos antes de DM

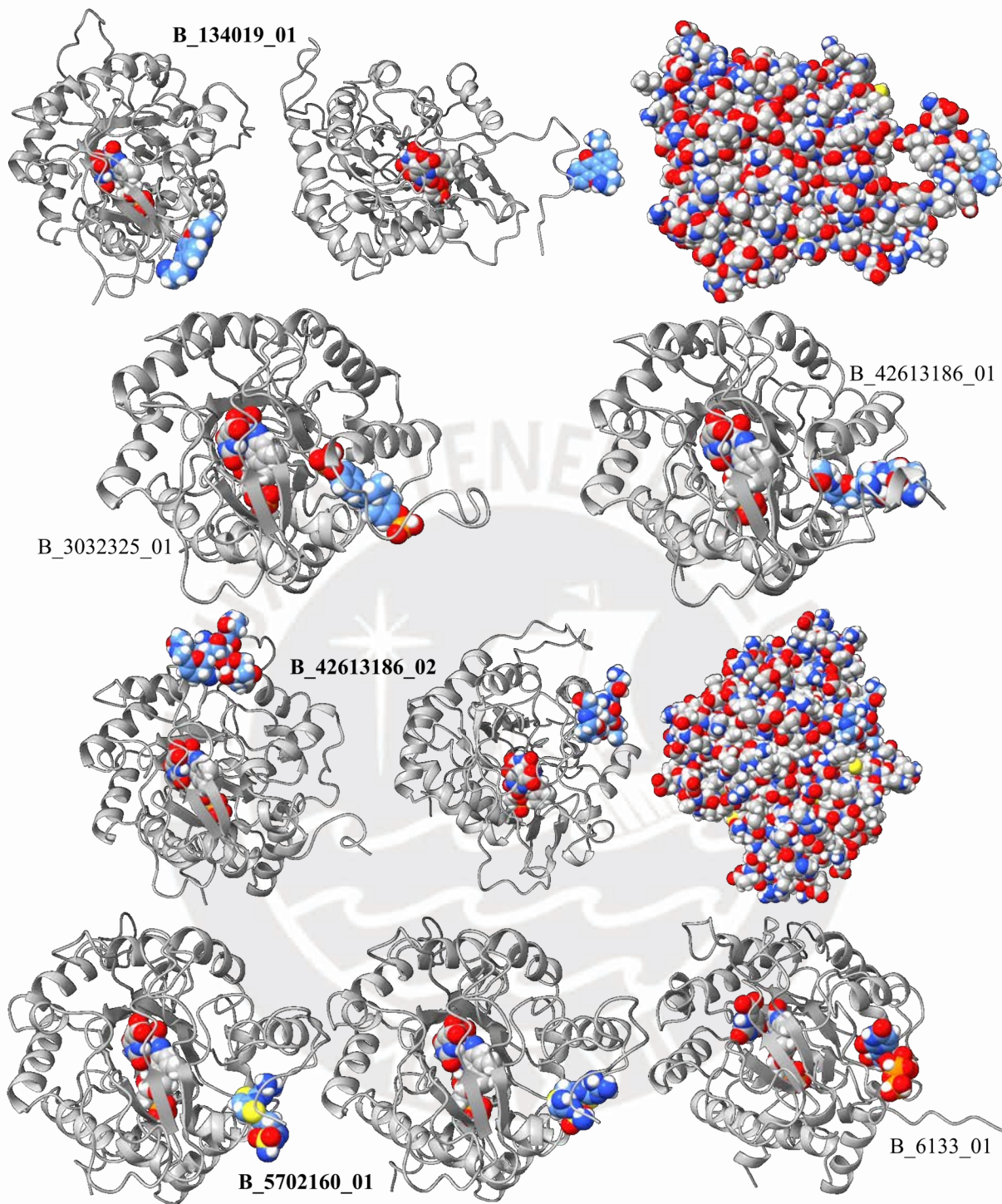


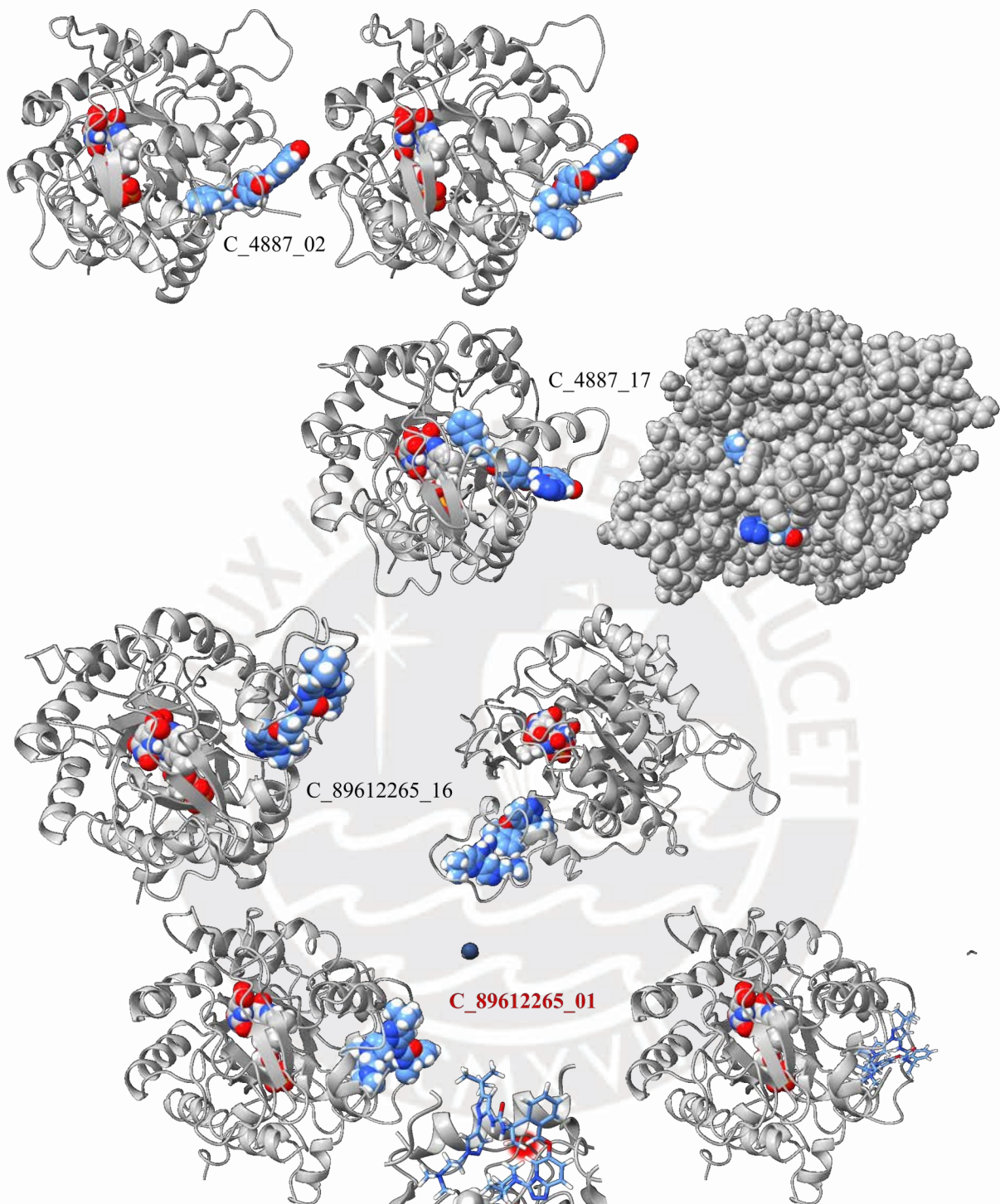


A.4.2. Posiciones finales

Se obtuvieron del último nano-segundo de cada simulación de DM.

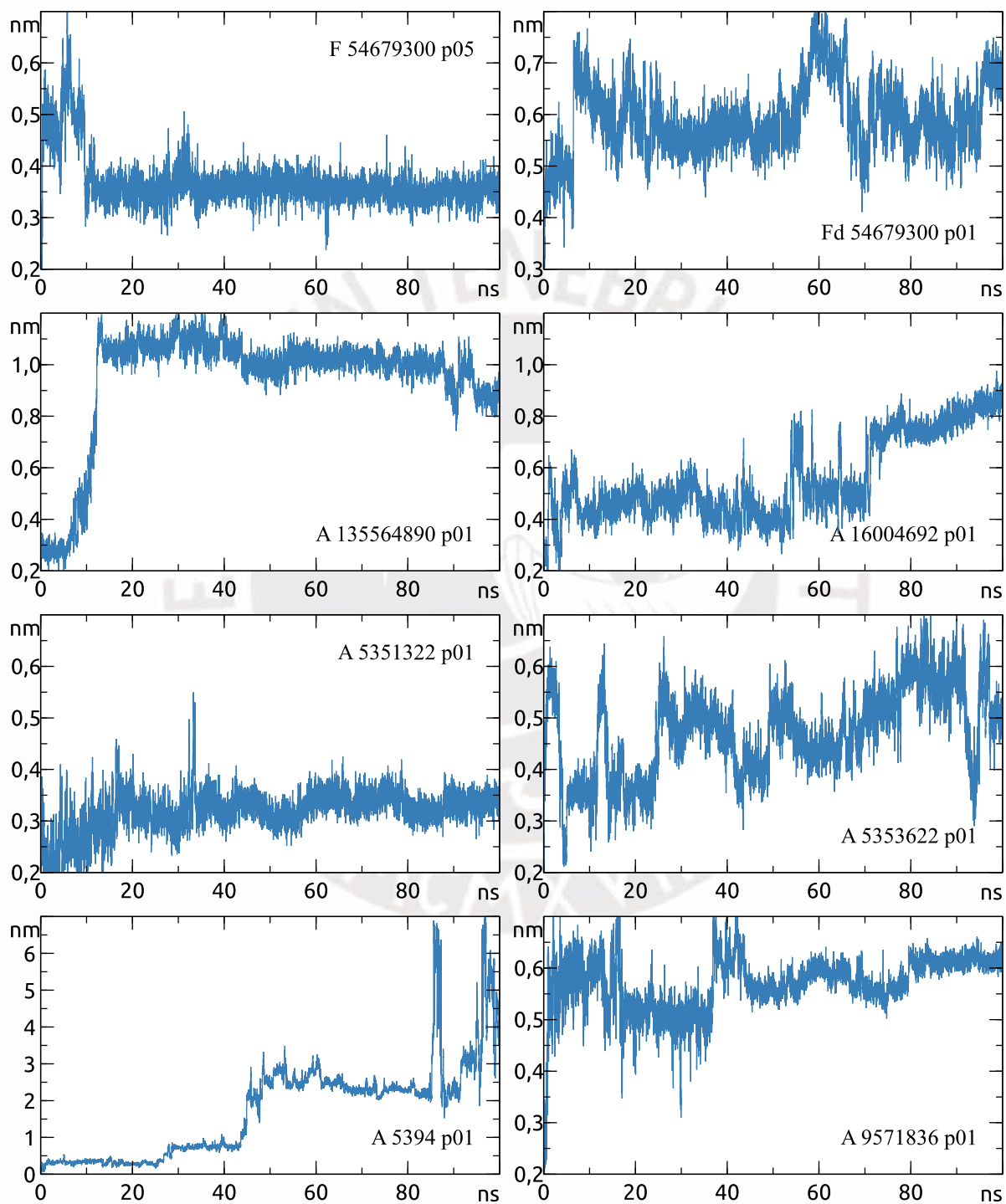


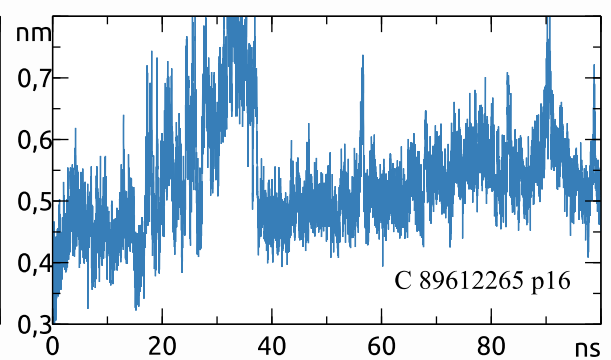
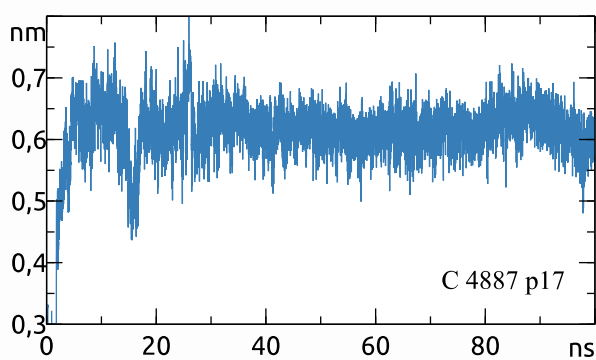
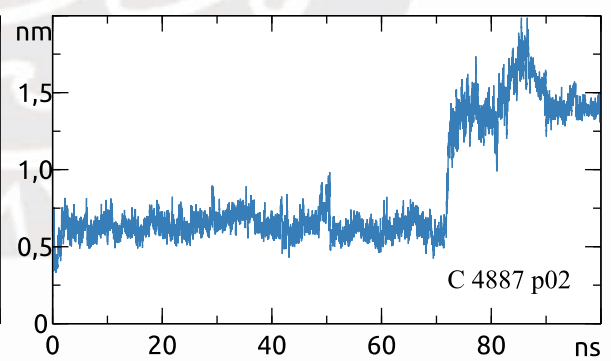
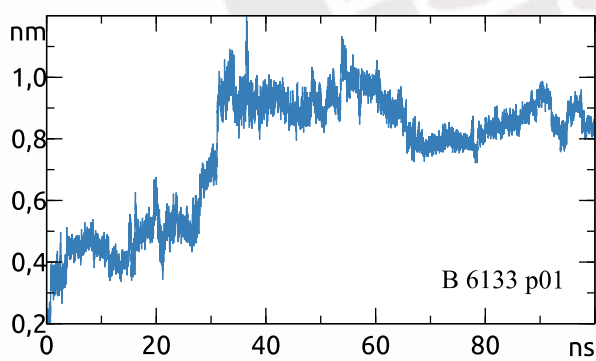
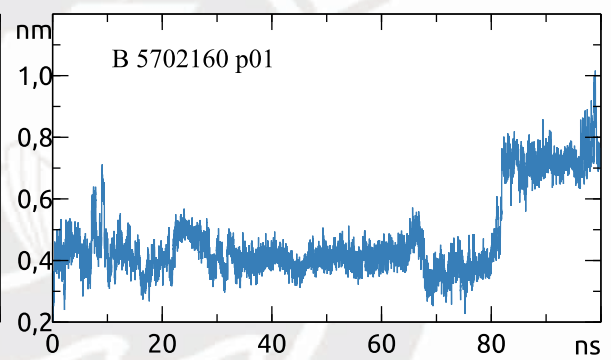
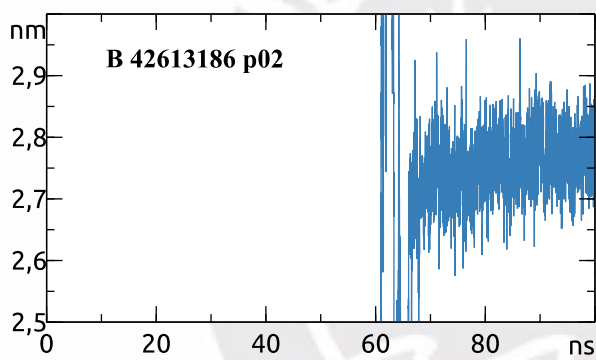
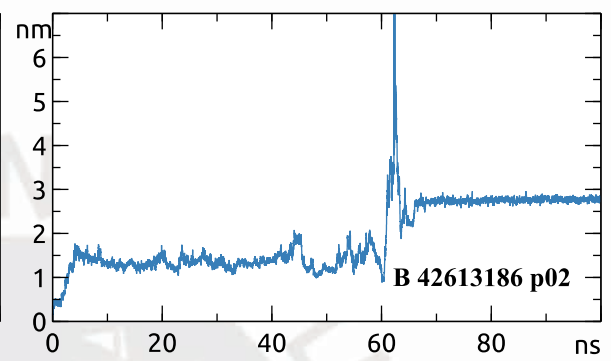
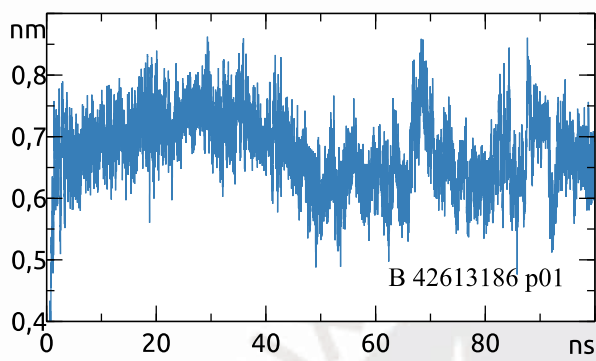
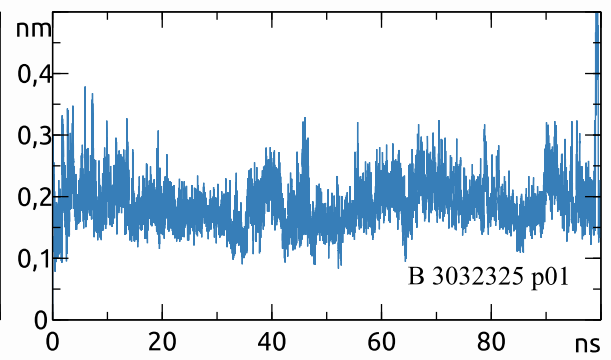
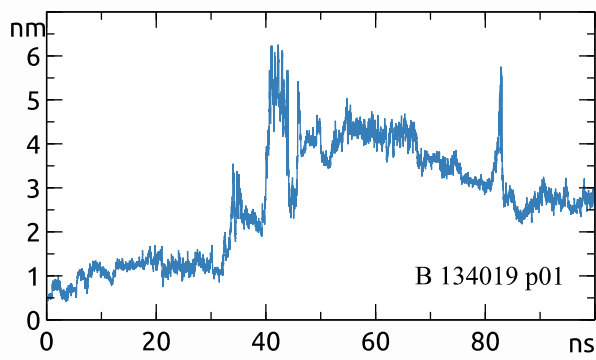




A.4.3. RMSD

La información para elaborar estas gráficas está en Resultados > DM > DM_rmsd_FeCof_100ns.csv.





A.4.4. RMSF

La información para elaborar estas gráficas está en Resultados > DM > DM_rmsf_99a100ns.csv y corresponden al último nano-segundo.

