TESIS

Doctorado en Informática

UNIVERSIDAD DE ALMERIA

Departamento de Informática

Computación de Altas Prestaciones para la Resolución de Problemas de Optimización en Bioinformática

(High Performance Computing for Optimization Problem Solving in Bioinformatics)

Almería, Octubre 2020

Autor:

Savíns Puertas Martín

Directores:

Juana López Redondo

Horacio Pérez Sánchez

Tesis Doctoral

Computación de Altas Prestaciones para la Resolución de Problemas de Optimización en Bioinformática

Universidad de Almería
Departamento de Informática
Doctorado en Informática (RD99/11)

Autor: Savíns Puertas Martín
Directores: Juana López Redondo
Horacio Pérez Sánchez

Almería, Octubre 2020

EDITADO CON LaTeX.

Plantilla original de Mathias Legrand (http://www.latextemplates.com).
Modificada por Savíns Puertas Martín.

Ph.D. Thesis

High Performance Computing for Optimization Problem Solving in Bioinformatics

University of Almería
Department of Informatics
Ph.D. in Informatics (RD99/11)

Author: Savíns Puertas Martín
Supervisors: Juana López Redondo
Horacio Pérez Sánchez

Almería, October 2020

A mis padres
y mis abuelos.

Agradecimientos (Acknowledgements)

Aquí me hallo escribiendo estas palabras, sin darme cuenta lo fugaces que han sido estos cuatro años. La felicidad hace corto el tiempo pero mi rostro, mi cuerpo y mi yo interior sabe que han sido cuatro años también muy duros, posiblemente los más difíciles de mi ya no tan corta vida. Pero no he estado solo en ningún momento. Hagamos un repaso.

En primer lugar, esto ha sido posible gracias a mis directores, Juani y Horacio. Aunque también me gustaría incluir en este grupo a Pilar pues no hay mesa coja de tres patas. Vosotros os habéis complementado fenomenalmente. Juani, siempre has estado pendiente de que se fueran cumpliendo los plazos de los trabajos y se avanzara. Si no funcionaba, ya estabas tú para hacerlo funcionar. Tu mensaje diario a las 10:15h de "Buenos días Savíns, ¿cómo vas?" ha ayudado sin ninguna duda. Horacio, eres tú y tus circunstancias. Siempre desde el punto de vista de hacer práctico y de interés el trabajo has permitido que todo lo realizado en esta tesis tenga una utilidad real y no sea abandonado en un cajón a su suerte. Por último, Pilar, has sabido gestionarlo todo tanto con mano izquierda como derecha además de aportar la experiencia adquirida por los distintos proyectos por los que has pasado.

También quiero agradecérselo a mis padres, que han sufrido todas mis particularidades. Creo que a partir de ahora, la situación será más sencilla. En estos cuatro años, en ocasiones no lo ha sido para nada pero hemos conseguido seguir adelante. Si algo me habéis enseñado es que lo que no nos mata, nos hace más fuertes.

Tampoco puedo olvidar del resto del grupo de investigación *TIC-146: Supercomputación y Algoritmos*, a los que están y los que estuvieron. Así, quiero darle las gracias a Inma, Leo, Ester, Eligius, Vicente, Martínez, Juan, Gloria, Nicolás, Francisco, Juanjo, Miriam, José Manuel, Cristóbal, Dani y Kostas. Entre ellos quiero destacar a dos. Por un lado a Miriam, con la que he compartido despacho desde que comencé y que siempre ha estado ahí. Y en segundo lugar a Juanjo, por toda la paciencia que ha tenido conmigo con el clúster y por el trabajo que le he dado manteniéndolo.

Nor can I forget what was possibly my happiest stage, except for the days when the accepted papers arrive. This was my time at the University of Kent at Canterbury. First of all, I would like to thank Professor Said Salhi for accepting me during that time. I also cannot forget all the people in our Kent Business School office. First of all, Mattia (Bevilacqua), who without his Italian welcome, it would not have been possible to have such close contact with the rest. That office had its charm. Rasmi (Jamil Rasmi Meqbel), Eirini (Bersimi), Sherri (Sherrihan Radi), Omar (Al-Bataineh), Ken (Cheng), Frankie (Yunlu Yang), Zhenhao (He), Orie (Miyazawa), Nunzia (Esposito), Rukiye (Kaya), Enoch (Nii Boi Quaye), Elmira (Partovi), Sim (Simdul Miri-Dashe), Siao-Leu (Phouratsamay), Prince (Ahmed Aljazea), Charikleia (Theodoraki), Nasser (AlShawaaf) and Iraklis (Apergis), thank you very much. And of course, my colleagues in Tile-Kiln, who helped me survive the first month and treated me phenomenally from start to finish, Hadi (al Hikmani), Orçun (Cetin), Çağrı (Aslan), Oguz (Önük), and Deming (Lin). I also want to mention Muradiye and especially my travel friend, Maria Bevz. None of us were English, but we all met there. I wish you all the best wherever you are.

También quiero agradecer a todas aquellas personas que de una u otra forma he conocido y han influido en mí durante estos cuatro años. Y entre todas ellas, quiero hacer mención a los

alumnos que he tenido durante mis tres años de docencia.

Para acabar, quiero mencionar todos los proyectos e instituciones que me han ayudado económicamente o mediante recursos en mi investigación permitiéndome asistir a congresos, enviar trabajos y realizar más experimentos. En primer lugar, mi beca de Formación de Profesorado Universitario financiada por el Ministerio de Educación, Cultura y Deporte (FPU15/02912) que sin ella no habría sido posible haber hecho el doctorado. Respecto a los proyectos, han sido varios: por parte de los Ministerios de Economía y Competitividad, los proyectos TIN2012-37483-C03-03, TIN2015-66680-C2-1-R, CTQ2017-87974-R y RTI2018-095993-B-100; por la Junta de Andalucía, los proyectos P10-TIC-6002, P11-TIC7176, P12-TIC301 y P18-RT-1193; por la Fundación Séneca–Agencia de Ciencia y Tecnología de la Región de Murcia, 19419/PI/14, 18946/JLI/13 y 20988/PI/18; por la Nils Coordinated Mobility, 012-ABEL-CM-2014A y por la UAL, UAL18-TIC-A020-B, financiado parcialmente por el Fondo Europeo de Desarrollo Regional. En cuanto a equipos, dar gracias al NLHPC (Laboratorio Nacional de Computación de Alto Rendimiento de Chile), a la Plataforma Andaluza de Bioinformática de la Universidad de Málaga, al CETA–CIEMAT (Centro Extremeño de Tecnologías Avanzadas) que pertenece a CIEMAT (Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas). También al Research Council y el Centro de Supercomputación de UiT de Noruega y finalmente al Centro de Supercomputación de Poznan en Polonia.

Y por último me gustaría dar gracias a Dios por estar siempre conmigo y hacer mi vida de esta manera.

Scientific Contribution

Scientific Journals

1. S. Puertas-Martín, J. L. Redondo, P. M. Ortigosa, and H. Pérez-Sánchez. **OptiPharm: An evolutionary algorithm to compare shape similarity.** Scientific Reports, 9, 2019. DOI: 10.1038/s41598-018-37908-6. JCR (2019) = 3.998. Subject categories = Multidisciplinary Sciences: 17/71 (Q1).
2. A. J. Banegas-Luna, J. P. Cerón-Carrasco, S. Puertas-Martín, H. Pérez-Sánchez. **BRUSELAS: HPC Generic and Customizable Software Architecture for 3D Ligand-Based Virtual Screening of Large Molecular Databases.** Journal of Chemical Information and Modeling, 59(6), 2805-2817, 2019. DOI: 10.1021/acs.jcim.9b00279. JCR (2019) = 4.549. Subject categories = Chemistry, Medicinal: 11/61 (Q1); Chemistry, Multidisciplinary: 49/177 (Q2); Computer Science, Information Systems: 28/156 (Q1); Computer Science, Interdisciplinary Applications: 17/109 (Q1).
3. S. Puertas-Martín, A.J. Banegas-Luna, M. Paredes-Ramos, J. L. Redondo, P. M. Ortigosa, O. O. Brovarets and H. Pérez-Sánchez. **Is high performance computing a requirement for novel drug discovery and how will this impact academic efforts?** Expert Opinion on Drug Discovery, 2020. DOI: 10.1080/17460441.2020.1758664. JCR (2019) = 4.887. Subject categories = Pharmacololy & Pharmacy: 30/270 (Q1).
4. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, and P. M. Ortigosa. **Optimizing Electrostatic Similarity for Virtual Screening: A New Methodology.** Informatica, 2020. DOI: 10.15388/20-INFOR424. JCR (2019) = 3.312. Subject categories = Computer Science, Information Systems: 46/156 (Q2); Mathematics, Applied: 9/260 (Q1).

International Conferences

1. S. Puertas-Martín, H. Den-Haan, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Enhancing Molecular Shape Comparison by a Global Evolutionary Algorithm.** 4th International Work-Conference on Bioinformatics and Biomedical Engineering, University of Granada, Granada, Spain, 20-22 April 2016.
2. S. Puertas-Martín, J. L. Redondo, H. den-Haan, H. Pérez-Sánchez, P. M. Ortigosa. **Multiobjective Based Scoring Function for Ligand Based Virtual Screening.** Proceedings of the XIII Global Optimization Workshop, GOW'16, pp. 105–108. University of Minho, Braga, Portugal, 4-8 September 2016. ISBN: 978-989-20-6764-3
3. S. Puertas-Martín, M. R. Ferrández J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Enhancing Molecular Shape Comparison by a Parallel Global Evolutionary Algorithm.** Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2017, Vol. 3, pp. 1722–1728. Rota, Spain, 4-8 July 2017. ISBN: 978-84-617-8694-7
4. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Virtual screening in molecular shape by using an evolutionary algorithm.** Proceedings OLA'2018 International Workshop on Optimization and Learning: Challenges and Applications, pp. 63–64. Alicante, Spain, 26-28 February 2018.
5. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Optimizing electrostatic similarity using a global evolutionary algorithm.** Proceedings of 6th EUROPT Workshop on Advances in Continuous Optimization, p. 20. Almería, Spain, 12-13 July 2018.

6. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Multi-objective evolutionary algorithm for Evaluation of Shape and Electrostatic Similarity.** Proceedings of LeGO 2018 - Int. Workshop on Global Optimization, Vol. 1, pp. 1–4. Leiden, The Netherlands, 18-21 September 2018.
7. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Virtual screening in electrostatic potential using an evolutionary algorithm.** Proceedings of META'18: 7th International Conference on Metaheuristics and Nature Inspired Computing , pp. 207–209. Marrakech, Morocco, 27-31 October 2018.
8. S. Puertas-Martín, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Maximizing the electrostatic similarity in drug discovery through evolutionary algorithms.** 8th International Work-Conference on Bioinformatics and Biomedical Engineering, University of Granada, Granada, Spain, 30 September-2 October 2020.

National Conferences

1. S. Puertas-Martín, M. R. Ferrández Juana L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Algoritmo evolutivo global como herramienta de cribado virtual utilizando la forma molecular.** III Jornadas Doctorales de la Universidad de Murcia, pp. 236–240. Murcia, Spain, 30 mayo-1 junio 2017. ISBN: 978-84-608-9779-8.
2. S. Puertas-Martín, M. R. Ferrández, J. L. Redondo, H. Pérez-Sánchez, P. M. Ortigosa. **Cribado virtual mediante un algoritmo evolutivo global paralelo.** Actas de las Jornadas SARTECO 2017 , pp. 177–179. Cáceres, Spain, 19-22 September 2017. ISBN: 978-84-697-4835-0.
3. M. R. Ferrández, S. Puertas-Martín, J. L. Redondo, B. Ivorra, A. M. Ramos, P. M. Ortigosa, **Computación de alto rendimiento para optimizar tratamientos térmicos de alta presión en la industria alimenticia** In Avances en arquitectura y tecnología de computadores. Actas de las Jornadas SARTECO 2017, pp. 119-122, September 2017. ISBN: 978-84-697-4835-0.
4. S. Puertas-Martín, J. J. Moreno, F. J. Orts, N. C. Cruz, J. L. Redondo, E. M. Garzón y P. M. Ortigosa. **Simulación de un procesador ARM para la enseñanza de Estructura de Computadores.** Actas de las Jornadas SARTECO 2019, pp. 235–240. Cáceres, Spain, 18-20 September 2019. ISBN: 978-84-09-12127-4.

Other works

1. M. R. Ferrández, S. Puertas-Martín, J. L. Redondo, B. Ivorra, A. M. Ramos and P. M. Ortigosa. **High performance computing for the optimization of high-pressure thermal treatments in food industry.** The Journal of Supercomputing, 75, 1187-1202, 2018. DOI: 10.1007/s11227-018-2351-4. JCR (2018) = 2.157. Subject categories = Computer Science, Hardware & Architecture: 22/53 (Q2); Computer Science, Theory & Methods: 35/105 (Q2); Engineering, Electrical & Electronic: 132/266 (Q2).
2. M. R. Ferrández, S. Puertas-Martín, J. L. Redondo, B. Ivorra, A. M. Ramos, and P. M. Ortigosa. **High-Performance Computing for Optimizing High-Pressure Thermal Treatments in Food Processing.** Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2017, Vol. 3, pp. 862–869. Rota, Spain, 4-8 July 2017. ISBN: 978-84-617-8694-7

Software copyright

- OPTIPHARM: an innovative evolutionary algorithm for virtual screening. P. M. Ortigosa; J. L. Redondo; S. Puertas-Martín and H. Pérez-Sánchez. University of Almería and Universidad Católica San Antonio de Murcia. Application number: 201899900606752. Patent number: RTA-94-18. Registration Date: 16/02/2018. Concession date: 09/08/2018.

Education

1. J. J. Moreno, S. Puertas-Martín, F. Orts, N. C. Cruz, J. L. Redondo, E. Garzón, P. M. Ortigosa. **On simulating an ARM processor for teaching computer structure**. 12th annual International Conference of Education, Research and Innovation (ICERI 2018), 11-13 November 2019, Seville, Spain.
2. F. Orts, N. C. Cruz, S. Puertas-Martín, M. Ruiz-Ferrández, J. J. Moreno, C. Medina-López, P. M. Ortigosa, V. Ruíz, L. Casado, J. M. Salmeron, J. L. Redondo, G. E. Garzón, G. Ortega, R. Villegas. **Learning Quantum computation Through Simple Examples** 11th International Conference of Education, Research and Innovation (ICERI 2019), Sevilla, Spain, 12-14 November 2019.
3. V. González-Ruiz, G. Ortega, E. Garzón, N. Calvo-Cruz, J. L. Redondo, J. Salmerón, L. Casado, P. M. Ortigosa, C. Medina-López, J. J. Moreno, M. R. Ferrández, F. Orts, S. Puertas-Martín, T. Santamaría-López **Collaborative Project-Based Learning: An Experience** 11th International Conference on Education and New Learning Technologies (EDULEARN 2020), Palma, Spain, 1-3 July 2019.

Teaching and Education

Teaching

2017-2018 (58 h)

Grado en Ingeniería Informática

- Estructura y Tecnología de Computadores: Grupo Trabajo: 2h.
- Multiprocesadores: Grupo Docente: 18h.
- Periféricos e Interfaces: Grupo Trabajo: 19h.
- Tecnologías de Acceso a Red: Grupo Trabajo: 19h.

2018-2019 (58 h)

Grado en Ingeniería Informática

- Estructura y Tecnología de Computadores: Grupo Trabajo: 38h.
- Tecnologías de Acceso a Red: Grupo Trabajo: 19h.

2019-2020 (63 h)

Grado en Ingeniería Informática

- Periféricos e Interfaces: Grupo Trabajo: 19h.
- Tecnologías de Acceso a Red: Grupo Docente: 10, Grupo Trabajo: 31h.

Plan de Formación del Profesorado 2018-2020

- Overleaf: plataforma colaborativa para redacción en Latex, 3h.

Dirección Trabajo Fin de Grado

- SmartFridge: Reconocimiento de acciones en un frigorífico y toma de decisiones en un ambiente inteligente. García Rodríguez, Rafael Alejandro. Directores: Juana López Redondo y Savíns Puertas Martín.

Education

1. Ephorus: programa antiplagio. 2 horas. 5 octubre de 2016.
2. Formación en competencias para la preparación de las pruebas de nivel B2 (inglés). 60 horas. 10 noviembre 2016 - 15 de febrero de 2017.
3. Formación en competencias para la preparación de las pruebas de nivel B2 (inglés). 60 horas. 6 marzo 2017 - 5 de julio de 2017.
4. Habilidades comunicativas. 10 horas. 24-25 de abril y 2-3 de mayo de 2017.
5. Deep Learning con TensorFlow. Andrés Ortíz García. Jornadas SARTECO, Universidad de Málaga. 19-22 septiembre de 2017.
6. Normalización de autores: perfil Orcid, ResearchID, ScopusID, etc. 2 horas. 8 de noviem-

bre de 2017.

7. Publicación en abierto: autoarchivo en el repositorio institucional de la UAL. 2 horas. 29 de noviembre de 2017.
8. Formación en competencias para la preparación de las pruebas de nivel B2 (inglés). 120 horas. octubre 2017 - junio de 2018.
9. Uso de las plataformas ARM en enseñanzas de informática. 13 horas. 7 y 8 de febrero de 2018.
10. Full Stack Mean. 25 horas. 3 de abril al 19 de junio de 2018.
11. Google Cloud Platform Education Grant en el entorno universitario. 12 horas. 17 al 20 de diciembre de 2018.
12. Metaheuristic Summer School. Acireale-Catania, Italia. 21-25 julio de 2018.
13. Formación en competencias para la preparación de las pruebas de nivel C1 (inglés). 60 horas. Febrero - junio de 2020.

Prefacio

Desde que existen registros, diferentes enfermedades han asolado a los seres vivos. Además, estas situaciones tiene cada vez un mayor impacto, especialmente en seres humanos, pues el incremento constante de la población y la hiperconectividad permiten transmitir enfermedades a una velocidad cada vez mayor dificultando su control. La necesidad de encontrar medicinas rápidamente ha desplazado las técnicas tradicionales basadas en análisis y estudios en el laboratorio, en favor de nuevas metodologías de caracter multidisciplinar. Actualmente, en el desarrollo de nuevos fármacos intervienen varias áreas de conocimiento, como son la química, la física, las matemáticas, la informática, etc. En este contexto se trabaja con modelos matemáticos/biológicos de las moléculas, algoritmos de optimización y técnicas de computación avanzada. La idea subyacente se basa en obtener un subconjunto de moléculas con una alta probabilidad de actuación contra una enfermedad, de entre todas las que se encuentran en una base de datos. Ese subconjunto será el que se analice en el laboratorio en fases posteriores del desarrollo. Este *cribado virtual* puede acelerar, en mucho, el proceso de descubrimiento de nuevos medicamentos, lo cual es de especial interés cuando aparecen enfermedades nuevas con una alta letalidad, como ha ocurrido recientemente con el COVID-19.

En esta tesis titulada Computación de Altas Prestaciones para la Resolución de Problemas de Optimización en Bioinformatica se plantean nuevos paradigmas para la resolución y optimización de problemas relacionados con el proceso de desarrollo de fármacos. En este sentido, las aportaciones se realizan sobre las técnicas de cribado virtual. En concreto, el cribado virtual basado en ligandos donde se buscan aquellos compuestos con el mayor valor de similitud de forma y potencial electrostático utilizando enfoques mono y multiobjetivo. La reducción del espacio de búsqueda mediante procedimientos dinámicos, la combinación de algoritmos de optimización global con métodos de búsqueda local y la ejecución paralela son algunas de las técnicas utilizadas en esta tesis para optimizar de forma eficaz y eficiente los problemas abordados. El conjunto de estas técnicas ha permitido analizar bases de datos que contienen hasta millones de compuestos y ofrecer nuevas y diferentes predicciones respecto a las de la literatura. Así pues, se han encontrado nuevos compuestos químicos que de otra forma seguirían esperando ser encontrados.

La tesis se divide en seis capítulos. En el primero de ellos se revisan los conceptos básicos del cribado virtual, la optimización global mono y multiobjetivo y la técnicas de computación de altas prestaciones. Entrando en detalle en cada de sus partes, la sección 1.1 realiza una revisión del cribado virtual basado en ligandos incluyendo los modelos matemáticos de los descriptores y las bases de datos utilizadas. En al sección 1.2 se presentan las definiciones de optimización global y problemas mono y multiobjetivo y se define cómo medir la calidad de sus soluciones. En la sección 1.3 se presentan algunos métodos heurísticos para abordar los problemas de optimización, con especial interés en los algoritmos meméticos. Finalmente, en la sección 1.4 se describen algunas de las arquitecturas paralelas y modelos de programación distribuidos realizando especial énfasis en la paralelización en centros de supercomputación.

El segundo capítulo detalla todos los procedimientos y características del algoritmo memético evolutivo que se ha desarrollado en esta tesis y que se ha llamado OptiPharm. En primer lugar, la sección 2.1 está dedicada a los parámetros de optimización, que son comunes para los distintos problemas que se resuelven en capítulos posteriores. En ese sentido, se presentan los distintos parámetros, se definen sus límites y se explica el cálculo de estos últimos ya que los límites

son dinámicos y dependen de las moléculas de entrada. En la sección 2.2 se detallan todos los procedimientos del algoritmo OptiPharm.

Los siguientes dos capítulos se centran en los dos problemas relacionados con el cribado virtual basado en ligandos. En concreto, el capítulo 3 aborda el problema de la similitud de forma. En primer lugar, la sección 3.1 define el problema de optimización a resolver y describe los métodos existentes. La sección 3.2 detalla las distintas configuraciones de ejecución de OptiPharm y de WEGA, el algoritmo de referencia en la literatura. Posteriormente, en las siguientes secciones se realizan diferentes estudios computacionales. En este tipo de problemas, al no conocerse la solución óptima, se debe de comparar en base a los resultados ya existentes. En consecuencia, en la sección 3.3 se hace un estudio con la base de datos Maybridge para comprobar que OptiPharm encuentra los compuestos similares cuando la base de datos así lo permite. En la sección 3.4 se compara la calidad de OptiPharm con la de WEGA utilizando la base de datos FDA y eliminando los hidrógenos. No considerar los hidrógenos es una práctica común en los algoritmos de la literatura pero en OptiPharm existe la posibilidad de considerarlos o no. En la sección 3.5 se compara la capacidad de clasificación de OptiPharm con WEGA utilizando las bases de datos DUD y DUD-E. Estas bases de datos han sido especialmente diseñadas para evaluar la calidad de la clasificación de los algoritmos distinguiendo entre ligandos y señuelos. Como parte final de los resultados, en la sección 3.6 se realiza un análisis con OptiPharm en el que se compara la influencia de los átomos de hidrógeno en los cálculos. Finalmente la sección 3.7 recoge las conclusiones obtenidas en ese capítulo.

En el capítulo 4 el análisis y los experimentos se centran en la similitud de potencial electrostático de los compuestos y más concretamente en la comparación de la metodología de la literatura y una nueva propuesta que se ha diseñado en esta tesis. En la sección 4.1 se exponen las dos metodologías y el problema de optimización. Ambas metodologías difieren en el objetivo de optimización abordado. Si bien, en esta tesis se propone una optimización directa de la similitud del potencial electrostático, en la literatura se opta por una optimización de la similitud de forma y una posterior selección en base al potencial electrostático. Además, también diferimos en el número de compuestos que se debe seleccionar de una fase a otra. Posteriormente, en la sección 4.2 se detallan las características de los experimentos y las configuraciones de los algoritmos. En la sección 4.3 se realiza un estudio analizando los efectos del número de compuestos que la metodología tradicional considera desde la etapa de optimización de la similitud de forma a la selección del potencial electrostático. En la sección 4.4 se comparan los resultados entre ambas metodologías. El capítulo termina con la sección 4.5 que recoge las conclusiones obtenidas.

El último capítulo de resultados es el capítulo 5. Aquí se aborda el problema desde una perspectiva multiobjetivo. Para ello se abordan de forma simultánea los dos problemas estudiados en los capítulos 3 y 4. En primer lugar, en la sección 5.1 se define el problema de optimización multiobjetivo. Posteriormente, en la sección 5.2 se realiza un estudio preliminar para seleccionar, de entre un conjunto de algoritmos multiobjetivo, aquel que mejor se adapta al problema. Para ello se realiza una batería de experimentos con distintas configuraciones y se selecciona aquel que mejor promedio de hipervolumen y tiempo de ejecución obtiene. En la sección 5.3 se describe el procedimiento para seleccionar los mejores candidatos a partir del conjunto de frentes de Pareto. Además, se comparan los resultados obtenidos con los algoritmos monoobjetivo y multiobjetivo. Los experimentos se realizan sobre la base de datos DrugBank. Por último, la sección 5.4 resume las conclusiones obtenidas del capítulo, destacando las ventajas de los algoritmos multiobjetivo relacionadas con la optimización simultánea de más de un descriptor.

Finalmente, el capítulo 6 recoge las conclusiones generales de esta tesis y marca las líneas de trabajo futuro.

Preface

Since records began, different diseases have plagued living beings. In addition, these situations are beginning to have an increasing impact, especially on humans, since the constant increase in population and hyperconnectivity allow diseases to be transmitted at an ever-increasing speed, making it difficult to control them. The need to find medicines quickly has displaced the traditional techniques based on analysis and studies in the laboratory, in favor of new methodologies of a multidisciplinary nature. Currently, several areas of knowledge are involved in the development of new drugs, such as chemistry, physics, mathematics and computer science, among others. In this regard, we are working with mathematical/biological models of molecules, optimization algorithms and advanced computing techniques. The underlying idea is to obtain a subset of molecules that will have a high probability of action against a disease, from all those found in a database. That subset will then be analyzed in the laboratory in later phases of development. This virtual screening can greatly accelerate the process of discovering new drugs, which is of particular interest when new diseases with a high level of lethality appear, as has recently been the case with COVID-19.

In this thesis entitled High Performance Computing for Optimization Problem Solving in Bioinformatics, new paradigms for the solution and optimization of problems related to the drug development process are proposed. Accordingly, contributions are made in terms of virtual screening techniques, more specifically, virtual screening based on ligands where we look for those compounds with the highest value of shape and electrostatic potential similarities using mono and multiobjective approaches. The reduction of the search space through dynamic procedures, the combination of global optimization algorithms with local search methods and parallel execution are some of the techniques used in this thesis to effectively and efficiently optimize the problems addressed. All these techniques have allowed the analysis of databases containing up to millions of compounds and thus offer new and different predictions to those in the literature. As a result, new chemical compounds have been found that otherwise would still be undiscovered.

The thesis is divided into six chapters. In the first one, the basic concepts of virtual screening, global monoobjetive and multiobjective optimization as well as high performance computing techniques are reviewed. Going into detail in each of its parts, section 1.1 carries out a review of ligand-based virtual screening including the mathematical models of the descriptors and databases used. In section 1.2, the definitions of global optimization along with single and multiobjective problems are described and defined as a measure of the quality of their solutions. In section 1.3, some heuristic methods to address optimization problems are presented, with special interest in memetic algorithms. Finally, in section 1.4, some of the parallel architectures and distributed programming models are described, with special emphasis on parallelization in supercomputing centers.

The second chapter details all the procedures and characteristics of the evolutionary memetic algorithm that has been developed in this thesis, called OptiPharm. First, section 2.1 is dedicated to the optimization parameters, which are common to the various problems that are solved in later chapters. Accordingly , the different parameters are presented, their limits are defined and the calculation of these parameters is explained, since the limits are dynamic and depend on input molecules. Finally, in section 2.2, all the procedures of the OptiPharm algorithm are detailed.

The next two chapters focus on the two problems related to ligand-based virtual screening. In particular, chapter 3 addresses the problem of shape similarity. Firstly, section 3.1 defines the optimization problem to be solved and describes the existing methods. Section 3.2 details the various execution configurations for OptiPharm and WEGA, the reference algorithm in the literature. Subsequently, in the following sections different computer studies are carried out. Since the optimal solutions to this type of problem are unknown, the comparisons of results will be made with the own results that we obtain. Consequently, in section 3.3, a study is performed with the Maybridge database to verify that OptiPharm is able to find similar compounds when the database allows it. In section 3.4, OptiPharm quality is compared with WEGA quality using the FDA database without considering the hydrogens. It is a common practice for algorithms in the literature but in OptiPharm we can choose whether to consider them or not. In section 3.5, the classification capability of OptiPharm is compared with WEGA using the DUD and DUD-E databases. These databases have been specially designed to evaluate the quality of the classification algorithms by distinguishing between ligands and decoys. As a final part of generating the results, an analysis with OptiPharm is carried out in section 3.6 comparing it with the influence of hydrogen atoms in the calculations. Finally, section 3.7 contains the conclusions obtained in this chapter.

In chapter 4, analysis and experiments are focused on the similarity of electrostatic potential of the compounds and more specifically, on the study of the methodology of the literature and a new proposal that has been designed in this thesis. In section 4.1, the two methodologies and the optimization problem are presented. Both methodologies differ from the optimization objective addressed. Although in this thesis a direct optimization of the electrostatic potential similarity is proposed, in the literature an optimization of shape similarity and a subsequent selection based on electrostatic potential is chosen. Furthermore, our work also differs regarding the number of compounds to be selected from one phase to another. Subsequently, in section 4.2, the characteristics of the experiments and the configurations of the algorithms are detailed. In section 4.3, a study is carried out analyzing the effects of the number of compounds that traditional methodology considers from the stage of optimization of shape similarity to the selection of electrostatic potential. After that, in section 4.4, the results between both methodologies are compared. The chapter ends with section 4.5 which brings together the conclusions obtained.

The last chapter of results is chapter 5. Here the problem is approached from a multiobjective perspective. To this end, the two problems studied in chapters 3 and 4 are tackled simultaneously. Firstly, in section 5.1, the multiobjective optimization problem is defined, as well as its restrictions. Then, in section 5.2, a preliminary study is carried out to select, from a set of multiobjective algorithms, the one that best suits the problem. This is done by performing a battery of experiments with different configurations and selecting the best average for hypervolume and execution time. In section 5.3, the procedure to select the best candidates from the set of Pareto fronts is described. In addition, the results obtained with the monoobjective and multiobjective algorithms are compared. The experiments are performed using the DrugBank database. Finally, section 5.4 summarizes the conclusions obtained from the chapter, highlighting the advantages of multiobjective algorithms related to the simultaneous optimization of more than one descriptor.

Lastly, chapter 6 gathers the general conclusions of this thesis and defines the lines for future work.

Abreviaciones

AUC *Area Under the Curve ROC (Receiver Operating Characteristic)* (Área bajo la curva ROC (Característica Operativa del Receptor)).

B&B *Branch and Bound* (Ramificación y Poda).

DUD *Directory of Useful Decoys* (Directorio de Señuelos Útiles).
DUD-E *Directory of Useful Decoys: Enhanced* (Directorio de Señuelos Útiles Mejorado).

EA *Evolutionary Algorithm* (Algoritmo Evolutivo).
EC *Evolutionary Computation* (Computación Evolutiva).

FDA *the Food and Drug Administration* (Administración de Alimentos y Medicamentos de Estados Unidos).

HPC *High Performance Computing* (Computación de Alto Rendimiento).
HTS *High Throughtput Screening* (Cribado de Alto Rendimiento).

LBVS *Ligand-Based Virtual Screening* (Cribado Virtual Basado en Ligandos).

MA *Memetic Algorithm* (Algoritmo Memético).
MIMD *Multiple Instruction, Multiple Data stream* (Flujo de instrucciones múltiple, Flujo de datos múltiple).
MISD *Multiple Instruction, Single Data stream* (Flujo de instrucciones múltiple, Flujo de datos único).
MOP *Multiobjective Optimization Problem* (Problema de Optimización Multiobjetivo).
MT *MultiThreadings* (Multihilo).

NMR *Nuclear Magnetic Resonance* (Resonancia Magnética Nuclear).
NUMA *Non-Uniform Memory Access* (Acceso a Memoria No Uniforme).

PBM *Population Based Metaheuristic* (Metaheurística Basada en Poblaciones).

PCA *Principal Component Analysis* (Análisis Principal de Componentes).
PFA *Pareto Front Approximation* (Aproximación del Frente de Pareto).

QSAR *Quantitative Structure-Activity Relationship* (Relación Estructura-Actividad Cuantitativa).

RS *Reference Set* (Conjunto de Referencia).

SASS *Single Agent Stochastic Search* (Búsqueda Estocástica de Agente Único).
SBVS *Structure-Based Virtual Screening* (Cribado Virtual Basado en Estructuras).
SD *Standard Deviation* (Desviación Estándar).
SIMD *Single Instruction, Multiple Data stream* (Flujo de instrucciones único, Flujo de datos múltiple).
SISD *Single Instruction, Single Data stream* (Flujo de instrucciones único, Flujo de datos único).
SPS *Set of Pareto-set approximation* (Aproximación del Conjunto de Conjuntos de Pareto).

UMA *Uniform Memory Access* (Acceso Uniforme a Memoria).

VS *Virtual Screening* (Cribado Virtual).

Índice general

Motivación y conceptos

1. Introducción

1.1 Cribado Virtual

Actualmente se llama COVID-19 [1], pero podría ser Ébola [2] o Zika [3]. Cada vez con más frecuencia, la aparición de nuevas enfermedades, además de las ya existentes, sitúan el proceso de desarrollo de fármacos en el primer plano del panorama mundial. Este no es un proceso sencillo, pues en la actualidad, la industria farmacéutica requiere de una media de 12 a 20 años y unos 850 millones de euros aproximadamente para el desarrollo y lanzamiento de un nuevo fármaco al mercado, lo cual supone un gran gasto de tiempo y dinero [4, 5]. En la figura 1.1 se pueden observar de manera resumida todas las etapas que forman el proceso de desarrollo de un fármaco comenzando con la identificación de una nueva enfermedad hasta su finalización con la aprobación y comercialización del medicamento.

El proceso comienza con la investigación sobre las causas de una enfermedad, que en algunos casos puede llevar a la identificación de uno o varios blancos moleculares asociados a la misma. Los pasos siguientes consisten en la identificación de compuestos activos con el blanco molecular y la optimización de su actividad biológica. Estos ensayos se hacen in-vitro con blancos moleculares aislados de las células. Los compuestos activos se someten a varias evaluaciones experimentales que implican ensayos a nivel celular, en animales y finalmente pruebas clínicas en humanos. Los compuestos que pasan satisfactoriamente por todas las etapas son aprobados para uso clínico por un agente regulatorio como puede ser la Agencia Española de Medicamentos y Productos Sanitarios[1] en España o la Administración de Alimentos y Medicamentos de Estados Unidos (FDA, *the Food and Drug Administration*)[2].

Uno de los principales problemas en el desarrollo de fármacos y por el que tanto recursos son necesarios es la gran cantidad de compuestos moleculares que se tienen que analizar. En

[1] `https://www.aemps.gob.es/`
[2] `https://www.fda.gov/`

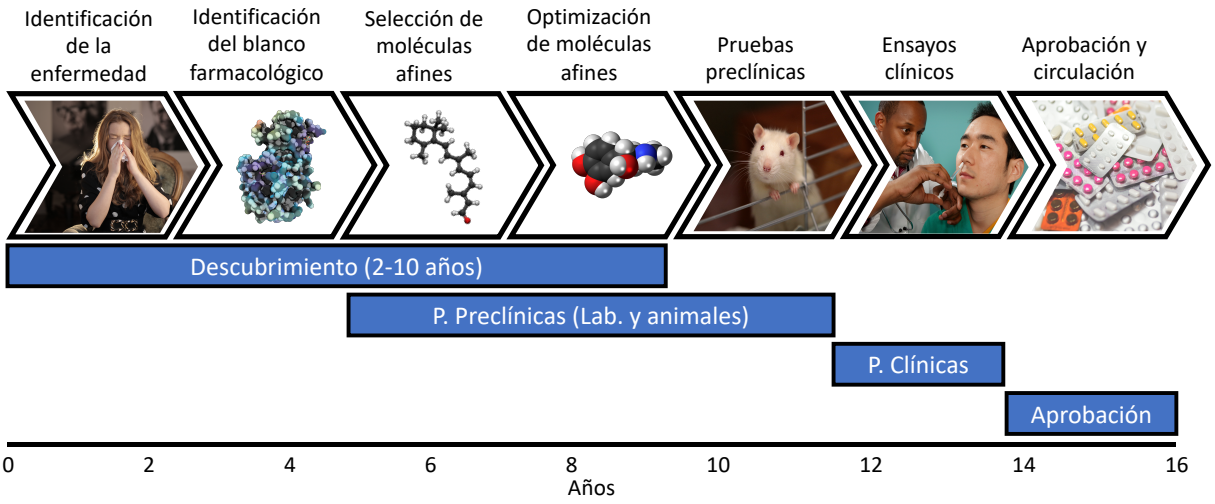


Figura 1.1: Etapas para el desarrollo de un nuevo fármaco.

2003 ya existían estudios que consideraban que la cantidad de compuestos orgánicos que son sintéticamente factibles se encontraba entre 10^{20} y 10^{24} [6]. En un trabajo más reciente de 2013, el número de compuestos se estimaba entre 10^{40} y 10^{200} [7]. Estas cantidades dejan en clara evidencia las limitaciones de las metodologías tradicionales y es por ello que en los últimos 30 años se ha incrementado la aparición y uso de metodologías apoyadas en los avances tecnológicos. Todas estas nuevas metodologías tienen como fin aplicarse en las etapas de identificación de blancos farmacológicos, selección de moléculas afines y optimización de estas que se corresponden con las etapas 2, 3 y 4 del proceso de desarrollo de un fármaco ilustrado en la figura 1.1. Dicho de otro modo, el objetivo de estas nuevas metodologías no es otro que reducir el número de moléculas que se probarán en etapas posteriores con la esperanza de que alguna de las seleccionadas demuestre su efectividad en los ensayos clínicos. De esta manera se reduce el tiempo de experimentación a la vez que se reduce el equipo material y humano necesario.

Respecto a las técnicas in-silico, una de las más desarrolladas y ampliamente utilizada es el Cribado Virtual (VS, *Virtual Screening*) o *in silico screening*. El término VS apareció por primera vez en la literatura científica en 1997 [8]. Se trata de una técnica computacional utilizada para identificar aquellos compuestos que son más afines o tienen características similares a uno dado de referencia. Al igual que los exámenes de Cribado de Alto Rendimiento (HTS, *High Throughtput Screening*), los métodos de VS normalmente se utilizan como un paso inicial en el proceso de descubrimiento de fármacos para enriquecer la fracción superior de la lista resultante de candidatos, que posteriormente será probada en ensayos clínicos [9]. La ventaja de VS con respecto a HTS es que VS permite procesar miles de compuestos en cuestión de horas y reducir la cantidad de compuestos que se sintetizarán, comprarán y/o probarán, disminuyendo los costes.

El VS consiste en un conjunto de etapas aplicadas secuencialmente donde se va reduciendo el número de compuestos a estudiar mediante la aplicación de diferentes métodos, que actúan como filtros para los compuestos que no interesan (ver figura 1.2). Esto permite aprovechar las fortalezas y evitar las limitaciones de los métodos individuales [9, 10]. Los compuestos que sobreviven a todos los filtros del VS generalmente se denominan compuestos candidatos y deben probarse experimentalmente en el laboratorio para confirmar su actividad biológica. Los métodos de VS se pueden clasificar en dos grupos principales:

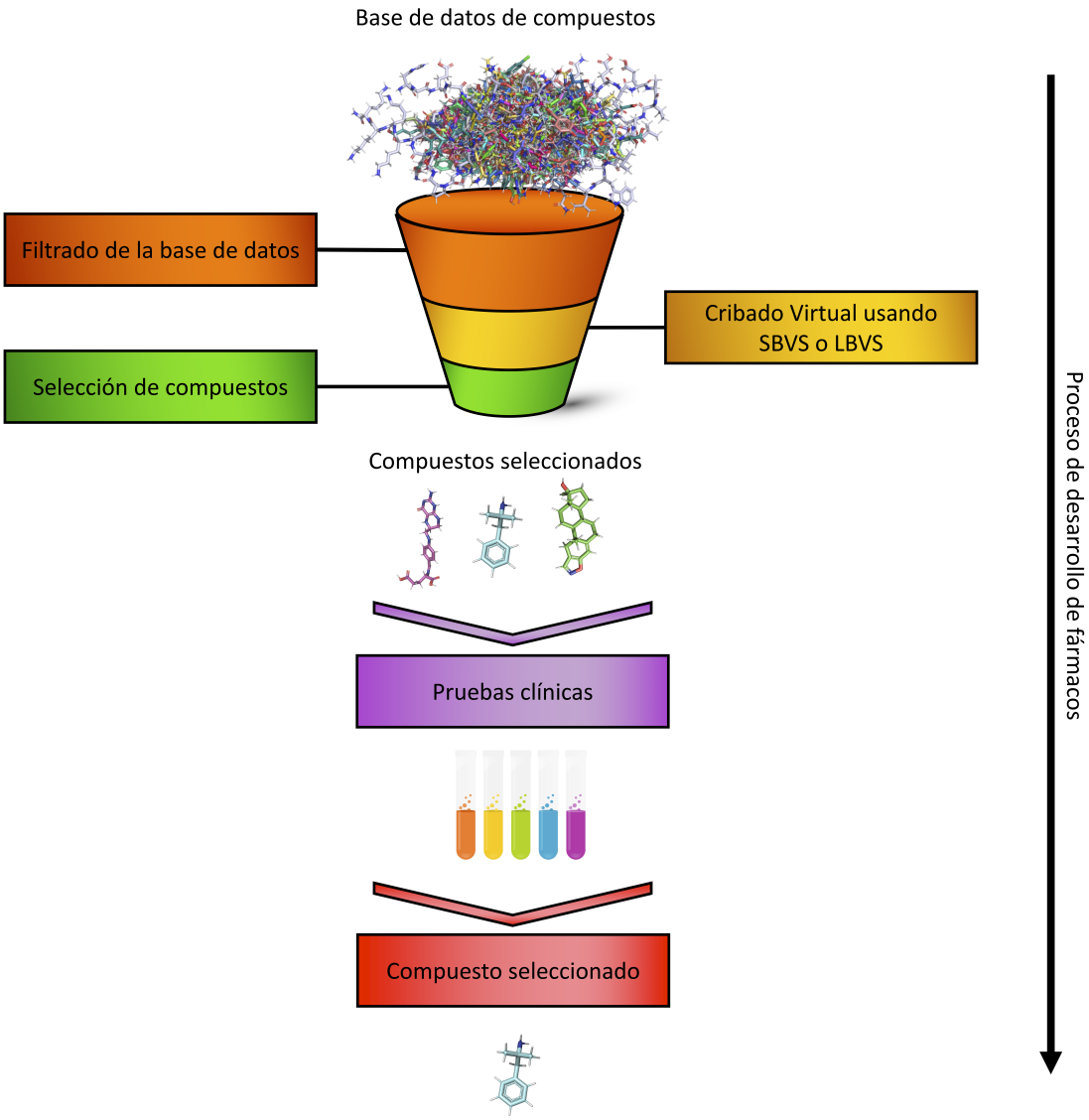


Figura 1.2: Representación esquemática del proceso de cribado virtual.

1. Cribado Virtual Basado en Estructuras (SBVS, *Structure-Based Virtual Screening*): se aplica cuando se dispone de la estructura tridimensional de la diana terapéutica, bien obtenida por métodos experimentales (cristalografía de rayos X [11] o Resonancia Magnética Nuclear (NMR, *Nuclear Magnetic Resonance*) [12]) o bien a través de la construcción de modelos moleculares. Un ejemplo de este tipo de VS es el docking [13-15], técnica en la que se intenta encontrar el mejor acoplamiento entre dos moléculas: un receptor y un ligando.
2. Cribado Virtual Basado en Ligandos (LBVS, *Ligand-Based Virtual Screening*): se utiliza cuando no se dispone de la estructura tridimensional del blanco farmacológico. Consiste en el análisis y comparación de diversos descriptores moleculares y datos de afinidad con respecto a ligandos conocidos, sin tener en cuenta la estructura de dicho receptor. Aquí se incluyen las técnicas de búsqueda de similitud mediante descriptores 2D/3D, Relación Estructura-Actividad Cuantitativa (QSAR, *Quantitative Structure-Activity Relationship*) [16] y técnicas de shape matching (comparación de la forma global o parcial entre moléculas) [17].

En esta tesis nos centraremos en técnicas LBVS. Para una mayor comprensión de SBVS se recomienda la lectura de los trabajos [18-20].

1.1.1 Cribado Virtual Basado en Ligandos

Los métodos de LBVS utilizan la información de los ligandos conocidos en lugar de la estructura de la proteína objetivo. Esto permite utilizarlos cuando no se dispone o no se puede conocer la información de dicha proteína. La base de estos métodos es el principio de propiedad de similaridad introducido por Johnson y Maggiora [21], que establece que compuestos similares tienen propiedades similares. Dicho esto, es probable que los compuestos con alta similitud con los compuestos de referencia se comporten de manera similar y tengan efectos similares.

La comparación entre los compuestos se realiza mediante sus características o descriptores. Existen más de 3000 descriptores distintos [22] y pueden representar desde propiedades fisicoquímicas a propiedades electrónicas de las moléculas, entre otras. Para mayor detalle, en [23] se muestra una clasificación de los descriptores y los métodos más usados en la bibliografía. En esta tesis se han seleccionado los descriptores de similitud de forma y del potencial electrostático en el espacio 3D.

1.1.2 Descriptores

En esta sección se va a explicar como se mide la similitud en forma y del potencial electrostático dadas dos moléculas.

1.1.2.1 Similitud de forma

La similitud de la forma 3D (figura 1.3) se basa en la idea de que si dos moléculas son similares en forma, tienen más posibilidades de que compartan propiedades físicas y exhiban una actividad biológica similar [24, 25]. Siguiendo este enfoque, en LBVS los compuestos de la bases de datos se comparan con la forma 3D de compuestos activos conocidos, que se utilizan como referencia. En la literatura se pueden encontrar diferentes modelos. Los más utilizados son el modelo de esferas sólidas [26, 27] y el modelo gaussiano [25, 28] siendo este último una evolución del primero y por tanto el modelo seleccionado en esta tesis. En concreto, el modelo matemático que se ha implementado en los algoritmos desarrollados en esta tesis se describe a continuación.

El volumen de solapamiento entre una molécula A y una molécula B se expresa como una combinación lineal de funciones gaussianas ponderadas [29] (ver ecuación 1.1).

$$V_{AB}^{g} = \sum_{i \in A, j \in B} w_i w_j v_{ij}^{g} \tag{1.1}$$

donde w_i y w_j son los pesos asociados a los átomos i y j, respectivamente. Este peso se calcula

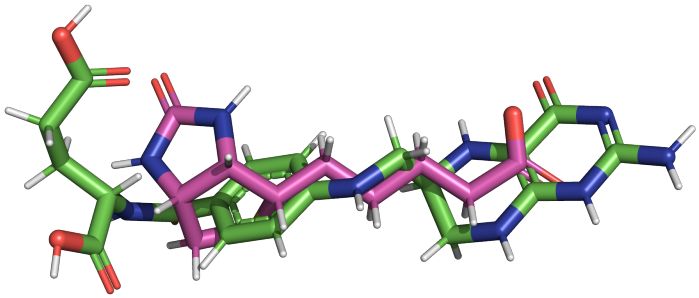


Figura 1.3: Ejemplo de dos compuestos superpuestos.

resolviendo la ecuación 1.2.

$$w_i = \frac{v_i^g}{v_i^g + k\sum_{j \neq i} v_{ij}^g} \tag{1.2}$$

donde k es una constante universal cuyo valor es 0.8665. v_i es el volumen del átomo i cuyo valor se calcula como $v_i = \frac{4\pi\sigma^3}{3}$.

Por último, el solapamiento de dos átomos v_{ij} se calcula como un producto de funciones gaussianas (ecuación 1.3).

$$v_{ij}^g = \int g_i(\boldsymbol{r}) g_j(\boldsymbol{r}) d\boldsymbol{r} = \int pe^{-(\frac{3p\pi^{\frac{1}{2}}}{4\sigma_i^3})^{\frac{2}{3}}(\boldsymbol{r}-\boldsymbol{r_i})^2} pe^{-(\frac{3p\pi^{\frac{1}{2}}}{4\sigma_j^3})^{\frac{2}{3}}(\boldsymbol{r}-\boldsymbol{r_j})^2} d\boldsymbol{r} \tag{1.3}$$

donde p es el parámetro que controla la suavidad de la esfera gaussiana, es decir, la altura de la función gaussiana y σ es el radio del átomo i.

Finalmente, hay que mencionar que el valor de solapamiento obtenido por la ecuación 1.1 no se encuentra acotado por ningún límite superior, mientras que su límite inferior es 0 (porque no existe solapamiento). El valor es más grande cuanto mayor sea el número de átomos de los compuestos pues el valor de solapamiento posiblemente sea mayor. Debido a esto, los resultados obtenidos con la ecuación 1.1 no puede compararse entre experimentos con distintos compuestos. Para poder hacerlo, es necesario realizar una normalización de los resultados de tal forma que sea igual para todos los compuestos. Para ello se utiliza la métrica de Tanimoto (ecuación 1.4).

$$Tc_S = \frac{V_{AB}^g}{V_{AA}^g + V_{BB}^g - V_{AB}^g} \tag{1.4}$$

donde V_{AA} y V_{BB} son el solapamiento de las moléculas A y B consigo mismas respectivamente. Esta función devuelve un valor comprendido en el rango $[0,1]$ donde 0 significa que no hay solapamiento y el valor 1 significa que existe una solapamiento completo entre ambas moléculas.

Este modelo explicado es el actual estado del arte para el cálculo de la similitud de forma de dos moléculas y por tanto el implementado en nuestros algoritmos. No obstante, a continuación se referencia una versión previa que sigue vigente en algunos software actuales y cuya diferencia

es la no consideración de los pesos de los átomos, w_i y w_j. El modelo se define en la ecuación 1.5.

$$V^g_{AB_R} = \sum_{i \in A, j \in B} v^g_{ij} \tag{1.5}$$

Igual que sucede con el primer modelo, existe la necesidad de normalizar los resultados para poder compararlos entre distintos compuestos con distinto tamaño. Para ello se utiliza nuevamente el coeficiente de Tanimoto 1.6.

$$Tc_R = \frac{V^g_{AB_R}}{V^g_{AA_R} + V^g_{BB_R} - V^g_{AB_R}} \tag{1.6}$$

1.1.2.2 Similitud de potencial electrostático

Las interacciones electrostáticas a menudo juegan un papel crítico en la unión del ligando debido a que el blanco farmacológico tiene un entorno electrostático particular que el ligando debe compensar de manera óptima para que se produzca la unión. Por lo tanto, utilizando el potencial electrostático del ligando como referencia podemos obtener compuestos que tienen una distribución electrostática similar y que potencialmente podrían coincidir con el entorno electrostático del objetivo, y en consecuencia, ser candidatos para tener una acción sobre ese objetivo.

Al igual que sucedía en el descriptor anterior, para el cálculo de la similitud del potencial electrostático también existen varios modelos. Principalmente se utilizan dos modelos, el basado en la Ley de Coulomb [30] y el de la ecuación de Poisson–Boltzmann [31]. En esta tesis utilizaremos el segundo por ser el ampliamente utilizado en la literatura.

Como se ha mecionado, este modelo consiste en resolver la ecuación de Poisson [31] (ecuación 1.7).

$$\nabla\{\varepsilon(r)\nabla\phi(r)\} = -\rho_{mol}(r) \tag{1.7}$$

donde $\phi(r)$ es el potencial electrostático, $\varepsilon(r)$ es una constante dieléctrica, y $\rho_{mol}(r)$ es la distribución de carga molecular. La similitud electrostática entre dos compuestos se obtiene calculando la ecuación 1.8.

$$E_{AB} = \int \phi^A(r)\phi^B(r)\Theta^A(r)\Theta^B(r)\mathbf{dr} \approx h^3 \sum_{ijk} \phi^A_{ijk}\phi^B_{ijk}\Theta^A_{ijk}\Theta^B_{ijk} \tag{1.8}$$

donde Θ es una función de enmascaramiento para asegurar que el potencial interior del compuesto no sea considerado en la propia comparación. Esta integral es la que se evalúa en una malla de puntos en tres dimensiones que contiene a los compuestos. El número de puntos viene definido por h. A mayor cantidad de puntos, más preciso será el cálculo pero también requerirá de una mayor cantidad de tiempo. En la figura 1.4 se ilustra un ejemplo de esta malla.

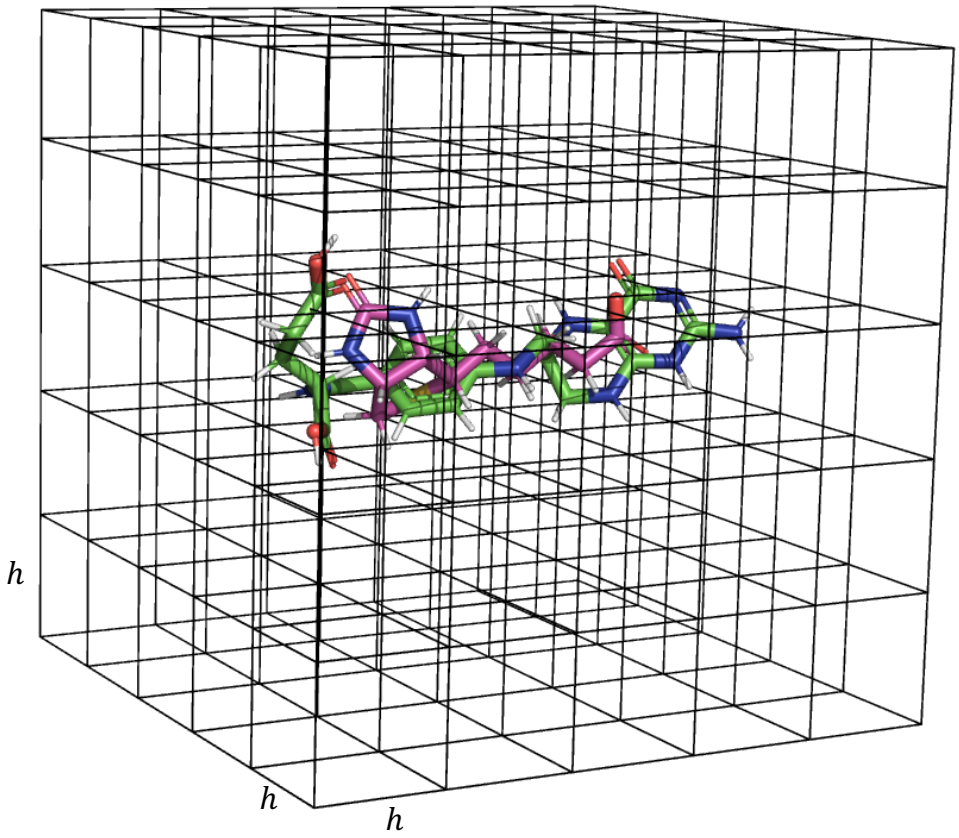


Figura 1.4: Ejemplo ilustrativo para el cálculo del potencial electrostático. En cada intersección se evalúa el potencial en dicho punto para cada molécula. Cuanto menor sea el valor de h, más preciso será el modelo pero también más costoso computacionalmente.

Al igual que sucede con la ecuación 1.1, el valor obtenido por la ecuación 1.8 depende del número de átomos de las moléculas comparadas. Para solucionar esto, y de la misma forma que se hizo con la función objetivo de la similitud de forma, se calcula la similitud de Tanimoto [32]:

$$Tc_E = \frac{E_{AB}}{E_{AA} + E_{BB} - E_{AB}} \tag{1.9}$$

donde E_{AB} representa el solapamiento del potencial electrostático de la molécula A con la molécula B. E_{AA} y E_{BB} es el solapamiento de las moléculas A y B consigo mismas, respectivamente. En este caso, la ecuación 1.9 tiene un valor en el rango $[-0.33, 1]$, donde -0.33 significa que los cambios de ambos compuestos tienen el mismo valor pero opuestas cargas, 0 significa que no hay solapamiento, y 1 significa que las cargas de las moléculas son las mismas.

1.1.2.3 Propiedades parametrizables

El cálculo de similitud de forma y del potencial electrostático dependen de las características de los compuestos evaluados. Dos de ellas son el número de átomos y el radio asignado a cada átomo, conocido como radio de Van der Waals. A continuación se muestran las configuraciones utilizadas en la literatura y las mejoras que proponemos.

(a) Con hidrógenos. (b) Sin hidrógenos.

Figura 1.5: Molécula de tirosina.

Átomos de hidrógeno

En la literatura es una práctica ampliamente extendida la eliminación de los hidrógenos para el cálculo de la similitud de forma. A modo ilustrativo, se muestra en la figura 1.5 una molécula de tirosina[3] con hidrógenos (figura 1.5a) y sin ellos (figura 1.5b). La razón por la que no se consideran es porque se acelera el cálculo de la similitud. Esta decisión se fundamenta en el hecho de que el volumen que aportan no es muy significativo (primer elemento de la tabla periódica y por tanto el de menor radio y volumen). Por su parte, en el cálculo del potencial electrostático sí se consideran los átomos de hidrógeno pues influyen considerablemente en los valores de similitud.

En esta tesis se ha analizado si omitir las moléculas de hidrógeno en los cálculos de similitud de forma es una buena práctica y si puede influir en la calidad de la predicción. Para ello, los algoritmos desarrollados permiten la inclusión o no de los átomos de hidrógeno. Como se verá en los capítulos de resultados, incluir los hidrógenos en el cálculo de similitud de forma mejora la calidad de las predicciones, lo cual es una aportación más a las contribuciones de esta tesis.

Radio de Van der Waals

Como se ha visto en la ecuación 1.1 utilizada para el cálculo de la similitud de forma, el radio considerado para cada elemento es uno de los parámetros que influye en el valor final (parámetro σ). Visualmente, las diferencias quedan reflejadas en la figura 1.6 donde se ha representado la misma molécula de morfolina[4] con las dos configuraciones. Si bien, en la literatura está ampliamente extendido el uso de un mismo radio para todos los elementos para reducir el tiempo de cálculo, en este tesis hemos considerado que podría ser más realista asignar un radio distinto a cada elemento. Por este motivo, los algoritmos que hemos diseñado permiten las dos opciones, diferenciándose nuevamente del resto de métodos de la literatura, donde se considera un radio de 1.7 Å para todos los elementos. En nuestro caso, nuestros algoritmos permiten ser configurados para asignar otros radios distintos mediante un archivo de configuración.

[3] `https://go.drugbank.com/drugs/DB00135`
[4] `https://go.drugbank.com/drugs/DB13669`

(a) Todos los elementos tienen el mismo radio. (b) Cada elemento tiene un radio diferente.

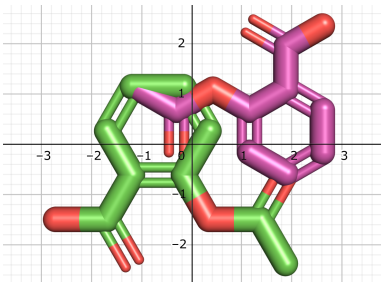
Figura 1.6: Molécula de morfolina con distintas configuraciones de radios de Van der Waals. En la subfigura (a) se puede observar que los átomos de hidrógeno (blanco), oxígeno (rojo), nitrógeno (azul) y carbono (verde) tienen el mismo tamaño. En la subfigura (b) la molécula tiene menos volumen pues cada átomo tiene su propio tamaño.

1.1.3 Bases de datos de moléculas

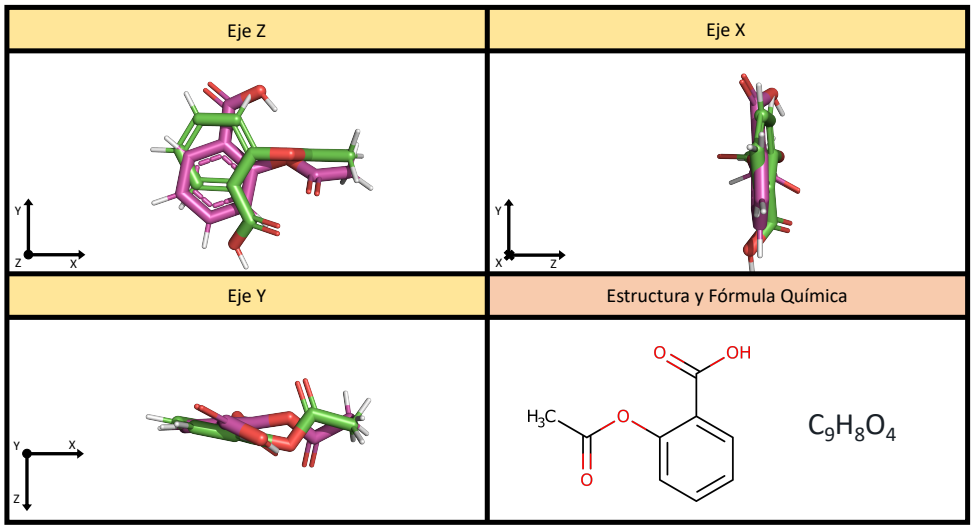
Para terminar esta sección falta hablar sobre la parte que realmente da sentido al cribado virtual y no es otra que las bases de datos de moléculas, las cuales están formadas por uno o varios archivos que contienen las características de los compuestos. Algunas de estas características son el nombre de los compuestos, la posición de sus átomos, la descripción de los enlaces, las cargas asociadas a cada átomo, etc. De este modo, con la lectura de los archivos de la base de datos de moléculas se obtienen los datos necesarios para utilizar los modelos matemáticos descritos en la sección 1.1.2 y para conocer el tipo de compuestos con los que se trabaja.

En el mundo empírico, una molécula no puede ser entendida como un elemento rígido, sino que dependiendo del estado o condiciones en las que se encuentre cambia sus características. Esto implica que la proximidad de sus átomos o los ángulos de los enlaces cambien, entre otras propiedades. Si bien se podrían documentar todas estas posibilidades para una molécula, estas serían un número exageradamente grande. Lo que ofrecen las bases de datos es una única configuración para todos los compuestos y después, dependiendo de las necesidades del problema, se le realizan unas modificaciones u otras. Dependiendo de su finalidad, cada base de datos ofrece una información diferente y su origen puede ser muy variado. Desde una colección interna de compuestos de interés a una base de datos como es ZINC [33], o un proveedor comercial de compuestos. Estos diferentes orígenes generan un problema para el VS. Por ejemplo, las coordenadas de los átomos se suelen dar indicando solo dos dimensiones en lugar de las tres dimensiones espaciales, y también habitualmente, no se proporciona información de los átomos de hidrógeno que hay en la molécula. En estos casos, se ha de realizar un post-procesamiento de lo que se ha leído en la base de datos de modo que se calcule la tercera dimensión o se incluyan los átomos de hidrógeno. Esta información suele ser necesaria para una gran cantidad de descriptores. Un ejemplo de estos problemas lo encontramos en la figura 1.7. En ella se pueden ver dos moléculas de aspirina[5] obtenidas de dos bases de datos diferentes, Chemspider [34] y DrugBank [35]. Se trata del mismo compuesto, pero el tamaño y la posición son completamente

[5] `https://go.drugbank.com/drugs/DB00945`



(a) Compuestos de aspirina sin procesar.



(b) Compuestos en 3D una vez procesados y alineados.

Figura 1.7: Moléculas de aspirina obtenidas de dos bases de datos diferentes. En la subfigura (a) se pueden observar los archivos sin modificaciones obtenidos de las bases de datos ChemSpider y Drugbank. En ella aparecen en 2D y sin hidrógenos. En la subfigura (b) se han procesado los compuestos asignando a cada compuesto su posición en la tercera dimensión además de añadirle los átomos de hidrógenos. Se pueden apreciar los cambios entre las dos moléculas del mismo compuesto.

distintos. Pero no solo la forma es dependiente de la fuente obtenida, sino que otros descriptores como la comparación de potencial electrostático, el acoplamiento de proteínas y ligandos o la detección de farmacóforos necesitan tener unos valores correctos en las cargas de los compuestos, ya que pueden no estar presentes, o pueden no estar correctamente asignados. Herramientas como OpenBabel [36] o FFMM94 [37] suelen ser útiles para dar un mismo formato a los compuestos y asignar correctamente las cargas. En definitiva, siempre que se obtiene una base de datos, esta debe ser previamente procesada.

1.1.3.1 Formato

Las bases de datos se encuentran representadas en muy diversos formatos dependiendo de la información que se necesite. Algunos de estos son *mol*, *mol2*, *sdf*, *pdb*, *xyz* o *gpr*. En esta tesis se han seleccionado el formato *mol2* y el *sdf*.

El formato *mol2* es un formato inventado originalmente por la empresa Tripos, Inc. Se

encuentra bien documentado y es ampliamente utilizado gracias en parte a que permite almacenar solo la información que se necesite evitando archivos de gran tamaño. La figura 1.8a muestra parcialmente la información de una molécula en dicho formato. En él se pueden diferenciar tres partes gracias a las etiquetas *MOLECULE*, *ATOM* y *BOND*. Cada apartado proporciona una información. *MOLECULE* ofrece los detalles generales de la molécula: nombre de la molécula según la base de datos de origen, número de átomos, número de enlaces, tamaño, etc. *ATOM* proporcina la información de cada átomo: nombre, coordenadas XYZ, etc. Y por último *BOND* índica los enlaces que existen y los átomos que lo forman. Existen más etiquetas que se pueden consultar en [38].

Sobre el formato *mol2* se ha realizado un exhaustivo trabajo considerando las diferentes partes de las que se compone el documento e incluyendo esa información dentro de los algoritmos desarrollados en esta tesis. Además, se han considerado todos los detalles técnicos para su correcto funcionamiento en sistemas operativos Windows, Linux y Unix.

Por otro lado, el segundo formato que se ha considerado es el *sdf*. La elección de este se debe a que es el formato de entrada para el software WEGA [29] con el que se ha comparado OptiPharm en el capítulo 3. El formato *sdf* es muy utilizado cuando solo se necesita información estructural de las moléculas, de ahí su nombre: *Structure-Data File*. En la figura 1.8b se muestra un ejemplo parcial de la misma molécula mostrada en la figura 1.8a donde se pueden ver las similitudes de algunos campos como el nombre, las coordenadas de los átomos y los enlaces.

1.1.3.2 Bases de datos utilizadas en esta tesis

Existe un gran número de bases de datos dependiendo de la finalidad de los experimentos. A continuación se detallan las bases de datos utilizadas en esta tesis y las modificaciones que se han realizado en cada una de ellas.

DrugBank

La base de datos DrugBank es una base de datos de libre acceso que se encuentra disponible online[6] y que contiene información de hasta 9891 fármacos en la versión 5.0.1 [39]. Además, contiene 4270 secuencias de proteínas no redundantes asociadas a dichos fármacos. Para cada compuesto, ofrece más de 200 campos de información y características asociadas al mismo.

En esta tesis hemos seleccionado un subconjunto de 1751 compuestos dentro de los 9891 totales, los cuales han sido aprobados por la Administración de Alimentos y Medicamentos de Estados Unidos (FDA, *the Food and Drug Administration*). La FDA es la Agencia Federal del Departamento de Salud y Servicios Humanos de los Estados Unidos responsable de proteger y promover la salud pública mediante el control, entre otras cosas, de medicamentos y de su comercio. Por tanto, descubrir nuevos compuestos con una alta similitud a alguno de ellos contribuiría directamente a la comercialización de dichos compuestos. Esto es destacable teniendo en cuenta la tendencia observada en los últimos años con el remplazo de medicamentos [24, 40, 41].

[6] `https://www.drugbank.ca`

```
@<TRIPOS>MOLECULE
DB01154
    35     35      0      0      0
SMALL
USER_CHARGES

@<TRIPOS>ATOM
      1 S1        4.0110    3.0907    0.5933 S.2     1 <0>    -0.3800
      2 O1        0.3914    0.3180    2.3367 O.2     1 <0>    -0.5700
     ...
     34 H17       4.0323   -3.0526    0.0368 H       1 <0>     0.1500
     35 H18       2.7808   -2.9310   -1.3303 H       1 <0>     0.1500
@<TRIPOS>BOND
     1    1   15 2
     2    2   11 2
    ...
    34   17   34 1
    35   17   35 1
```

(a) Formato *mol2*.

```
DB01154
 OpenBabel12221612453D

 35 35  0  0  1  0  0  0  0  0999 V2000
    4.0110    3.0907    0.5933 S   0  0  0  0  0  0  0  0  0  0  0  0
    0.3914    0.3180    2.3367 O   0  0  0  0  0  0  0  0  0  0  0  0
   ...
    4.0323   -3.0526    0.0368 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.7808   -2.9310   -1.3303 H   0  0  0  0  0  0  0  0  0  0  0  0
  1 15  2  0  0  0  0
  2 11  2  0  0  0  0
  ...
 17 34  1  0  0  0  0
 17 35  1  0  0  0  0
M  END
$$$$
```

(b) Formato *sdf*.

Figura 1.8: Un ejemplo de una molécula representada en dos formatos distintos.

Los compuestos seleccionados han sido procesados mediante la herramienta AmberTools [42] eliminando sales y neutralizando el estado de protonación, computando y asignando las cargas parciales del campo de fuerza MMFF94, agregando los átomos de hidrógeno y minimizando energías [43].

Maybridge Screening Hit Discovery

La base de datos Maybridge Screening Hit Discovery [44] con más de 53.000 compuestos es una biblioteca comercial formada por compuestos orgánicos pequeños con una alta similitud a blancos farmacológicos y compuestos experimentales [45], que cubre el 87 % de los 400.000 medicamentos farmacofóricos teóricos. Además cumple la regla de cinco de Lipinsky [46] y de buenas propiedades ADME(T) (Absorción, Distribución, Metabolismo, Excreción y (Toxicidad)) [47]. La colección HitCreatorTM (una selección de 14.400 compuestos filtrados de Maybridge) tiene como objetivo representar la diversidad de la colección principal que cubre el espacio químico de los medicamentos. Maybridge también ofrece una biblioteca de fragmentos (30.000 fragmentos), una colección de bloques de construcción a medida y una biblioteca de fragmentos de diversidad llamada Ro3 2500 (2500 fragmentos) con un índice de similitud de Tanimoto de

0.66 (basado en las huellas digitales estándar de Daylight[7]) asegurando la solubilidad de los compuestos y estando optimizados según SPR y Ro3. La base de datos original Maybridge se ha descargado de `https://www.maybridge.com`.

Directory of Useful Decoys

El Directorio de Señuelos Útiles (DUD, *Directory of Useful Decoys*) [48] es un benchmark mediante el cual los métodos de VS comprueban cómo de eficientes son diferenciando ligandos (o compuestos activos), que se sabe que se unen a un objetivo proteico dado, de señuelos. Los datos de entrada para cada molécula de cada conjunto contienen su estructura molecular e información sobre si está activo o no. La información sobre ligandos para cada proteína del conjunto DUD se tomó de datos experimentales. Los señuelos se prepararon para parecerse físicamente a los ligandos, pero al mismo tiempo, para ser químicamente diferentes a estos, por lo que es muy poco probable que actúen como agente aglutinante con la proteína. En promedio, para cada ligando es posible encontrar 36 moléculas señuelo que son muy similares en términos físicos, pero con una topología muy diferente. Los detalles sobre cómo se prepararon los señuelos (seleccionados de moléculas ya existentes en la base de datos ZINC [33]) se pueden encontrar en su trabajo original [48] por lo que solo mencionaremos aquí los detalles principales:

1. La base de datos inicial se creó utilizando 3.5 millones de moléculas compatibles con Lipinski de la base de datos ZINC de compuestos disponibles comercialmente (versión 6, diciembre de 2005).
2. Las características de las huellas digitales clave se calcularon utilizando las subestructuras clave de tipo 2 predeterminadas de CACTVS [49] y el análisis de similitud basado en huellas digitales se realizó con el programa SUBSET. Se seleccionaron los compuestos con valores de *Tc* inferiores a 0.9 para cualquier ligando anotado (llamados activos). Esto redujo el número de compuestos de ZINC a 1.5 millones de moléculas topológicamente diferentes a los ligandos.
3. El programa QikProp (Schrodinger, LLC, Nueva York, NY) se usó para calcular 32 propiedades físicas de todos los ligandos anotados y compuestos ZINC seleccionados del paso anterior, y se aplicó QikSim (Schrodinger, LLC, Nueva York, NY) para priorizar compuestos de ZINC que poseen propiedades físicas similares a cualquiera de los ligandos.
4. Se utilizó un peso de 4 para enfatizar los descriptores similares a los medicamentos (peso molecular, número de aceptores de enlaces de hidrógeno, número de donadores de enlaces de hidrógeno, número de enlaces rotativos y log P), mientras que el resto de los descriptores fueron ignorados (peso 0) durante el procedimiento de análisis de similitud.
5. Finalmente, se seleccionaron 36 compuestos señuelos para cada ligando, lo que condujo a un total de 95316 señuelos que fueron similares en términos de propiedades físicas pero topológicamente diferentes a los 2950 ligandos anotados. El número total de señuelos es menor de 36 veces el número de ligandos porque algunos ligandos tienen los mismos señuelos.

Para la tesis se ha utilizado la base de datos DUD original descargada de `http://zinc.docking.org`.

[7] `https://www.daylight.com/`

Database of Useful Decoys: Enhanced

El Directorio de Señuelos Útiles Mejorado (DUD-E, *Directory of Useful Decoys: Enhanced*) [50] es la nueva versión de DUD en la que se ha incrementado el número de blancos farmacológicos, ligandos y señuelos. Ha sido diseñada por el Laboratorio Shoichet en la UCSF. La metodología del benchmark DUD-E está descrita completamente en su trabajo original [50]. El benchmark consta de 102 blancos farmacológicos, 22886 ligandos (un promedio de 224 ligandos por blanco farmacológico) y 50 señuelos por ligando [51]. La base de datos DUD-E ha sido descargada de `http://dude.docking.org/`.

1.2 Optimización global mono y multiobjetivo

La optimización global se encuentra presente en numerosas áreas del conocimiento tales como la economía, la biología, la química, etc. Esto ha dado lugar a la aparición de gran cantidad de aplicaciones y estrategias. Toda ellas se pueden clasificar dentro de uno de los dos grandes subconjuntos: métodos exactos o métodos heurísticos. Uno u otro depende de si puede garantizar que se haya encontrado la solución óptima. Esta sección se centra en los algoritmos heurísticos que son los que se han desarrollado a lo largo de esta tesis.

1.2.1 Optimización global

El objetivo de la optimización global es encontrar la mejor solución (global) de un problema dado, en presencia de múltiples soluciones óptimas locales y globales. Formalmente, la optimización global busca la solución global de un problema de optimización delimitado por restricciones. Los problemas no lineales están omnipresentes en muchas aplicaciones como por ejemplo, diseños de ingeniería, biotecnología, análisis de datos o gestión ambiental, entre otras. Su solución a menudo requiere de un enfoque de búsqueda global. Para un mejor entendimiento, a continuación se definen los términos de optimización local y global.

Definición 1.2.1 — Óptimo local. Una función de valor real $f : S \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ tiene un mínimo local (resp. máximo local) en un punto $\mathbf{x}^* \in S$ si y solo si existe $\varepsilon > 0$ tal que $f(\mathbf{x}^*) \leq f(\mathbf{x})$ (resp. $f(\mathbf{x}^*) \geq f(\mathbf{x})$) para todo $\mathbf{x} \in S$ tal que $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$.

Definición 1.2.2 — Óptimo global. Una función de valor real $f : S \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ tiene un mínimo global (resp. máximo global) en un punto $\mathbf{x}^* \in S$ si y solo si satisface $f(\mathbf{x}^*) \leq f(\mathbf{x})$ (resp. $f(\mathbf{x}^*) \geq f(\mathbf{x})$) para todo $\mathbf{x} \in S$.

La representación matemática de un problema de optimización global, en su forma de maximización, viene dada por:

$$\begin{aligned} &\text{máx } f(x) \\ &\text{t. q. } x \in S \end{aligned} \tag{1.10}$$

donde S es un conjunto cerrado no vacío en $\mathbb{R}^n$ y f es una función continua. Por tanto, el objetivo es encontrar el valor máximo f^* en todos los puntos $x^* \in S$ tal que:

$$f^* = f(x^*) \geq f(x) \quad \forall x \in S \tag{1.11}$$

1.2.2 Optimización multiobjetivo

Para problemas multiobjetivo donde hay más de una función objetivo a optimizar, la formulación de los problemas monoobjetivo se amplía. En general, un Problema de Optimización Multiobjetivo (MOP, *Multiobjective Optimization Problem*) se expresa de la siguiente manera:

$$\begin{aligned} &\{\text{máx}\, f_1(\mathbf{x}), \ldots, \text{máx}\, f_m(\mathbf{x})\} \\ &\text{t.q.} \quad \mathbf{x} \in S \subseteq \mathbb{R}^n \end{aligned} \tag{1.12}$$

donde $S \subseteq \mathbb{R}^n$ es la región factible de los vectores de decisión $\mathbf{x} = (x_1, \ldots, x_n)$, con $n \in \mathbb{N}$, y $f_1, \ldots, f_m : \mathbb{R}^n \longrightarrow \mathbb{R}$ son las funciones objetivo. Los vectores imagen m-dimensionales $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ son referidos como vectores objetivo y la imagen de un conjunto factible en el espacio objetivo $\mathbb{R}^m$ es llamada región objetivo factible $Z = \mathbf{f}(S)$.

En esta tesis, tratamos con MOP continuos no lineales, debido a que las variables de decisión involucradas pertenecen a un rango continuo de valores reales y las funciones objetivo están dadas por modelos matemáticos no lineales.

A menudo, en un MOP, los diferentes objetivos se contradicen entre sí. Como consecuencia, no siempre existe una solución única que maximice todas las funciones objetivo al mismo tiempo. En lugar de ello, la solución consiste en varios puntos de compromiso del espacio factible.

A continuación, se introduce la terminología multiobjetivo clásica para formular teóricamente el concepto de solución de un MOP.

Definición 1.2.3 Para dos vectores factibles $\mathbf{x}, \mathbf{x}' \in S$, se dice que $\mathbf{x}$ domina $\mathbf{x}'$ y $\mathbf{f}(\mathbf{x})$ domina $\mathbf{f}(\mathbf{x}')$ si y solo si $f_i(\mathbf{x}) \geq f_i(\mathbf{x}')$ para todo $i = 1, \ldots, m$, y existe un $j \in \{1, \ldots, m\}$ tal que $f_j(\mathbf{x}) \geq f_j(\mathbf{x}')$.

Definición 1.2.4 Un vector de decisión $\mathbf{x} \in S$ se llama eficiente o solución de Pareto óptima si y solo si no existe otro vector factible $\mathbf{x}' \in S$ que domine $\mathbf{x}$, es decir, ninguna de las funciones objetivo puede mejorarse sin empeorar al menos una de las otras. El conjunto S_E de todas las soluciones de Pareto óptimas se llama conjunto eficiente o conjunto óptimo de Pareto. La imagen de una solución de Pareto óptima $\mathbf{f}(\mathbf{x})$ se llama vector objetivo de Pareto óptimo y el conjunto de todos los vectores objetivo de Pareto óptimos $\mathbf{f}(S_E)$ se denomina frente de Pareto óptimo.

La eficiencia se define en el espacio de decisión. La correspondiente definición en el espacio objetivo (criterio) es la siguiente.

Definición 1.2.5 Un vector objetivo $\mathbf{z}^* = f(\mathbf{x}^*) \in Z$ se llama no dominado si y solo si $\mathbf{x}^*$ es eficiente. El conjunto Z_N de todos los vectores no dominados es llamado conjunto no dominado o frente de Pareto. Si $\mathbf{x_1}$ y $\mathbf{x_2}$ son dos puntos factibles y $\mathbf{x_1}$ domina a $\mathbf{x_2}$, entonces se dice que $f(\mathbf{x_1})$ domina a $f(\mathbf{x_2})$.

Por tanto, resolver un MOP formulado como en la ecuación 1.12 significa encontrar todo el subconjunto no dominado formado por todos los vectores de decisión eficientes, cuyos correspondientes vectores objetivos representan el frente de Pareto óptimo [52].

Se puede establecer un límite superior e inferior para este frente de Pareto óptimo mediante los vectores objetivo de nadir e ideal, respectivamente, como se explica a continuación.

Definición 1.2.6 El vector objetivo de nadir $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_m^{\text{nad}})$ se define como el vector con el peor valor que las funciones objetivo pueden alcanzar en el frente de Pareto óptimo. Por el contrario, el *vector objetivo ideal* $\mathbf{z}^{\text{id}} = (z_1^{\text{id}}, \ldots, z_m^{\text{id}})$ se compone de los mejores valores que cada función objetivo puede alcanzar en el frente óptimo de Pareto. Por lo tanto, cada componente de estos vectores puede obtenerse como $z_i^{\text{nad}} = \text{mín}_{\mathbf{x} \in S_E} f_i(\mathbf{x})$, $z_i^{\text{id}} = \text{máx}_{\mathbf{x} \in S_E} f_i(\mathbf{x})$, para todo $i = 1, \ldots, m$, respectivamente.

Sin embargo, para la mayoría de los MOP es prácticamente imposible obtener una descripción exacta del conjunto eficiente (o frente de Pareto), ya que estos conjuntos suelen ser continuos e incluyen un número infinito de puntos. Además, el coste computacional puede ser excesivo, y este es un aspecto importante, sobre todo para los problemas de optimización que no se pueden resolver fácilmente, como los considerados en esta tesis. En consecuencia, los actuales algoritmos del estado del arte para resolver MOP se centran en proporcionar un conjunto finito de puntos que componen una Aproximación del Frente de Pareto (PFA, *Pareto Front Approximation*) como solución de la ecuación 1.12. Como se detalla en la siguiente subsección, la bondad de estos PFA se mide en términos de su cercanía al frente real de Pareto y su distribución uniforme para cubrir todo el frente.

1.2.3 Indicadores de calidad

La calidad y la eficacia de las PFA deben medirse para asegurar que son un conjunto discreto de puntos que conforman el frente de Pareto y están bien distribuidos sobre él. En esta tesis hemos utilizado el método del *indicador de calidad* para la evaluación y comparación de las aproximaciones de los conjuntos de Pareto (en [53] se puede obtener más información sobre este método y los otros existentes). Es el más extendido en la literatura y consiste en asignar a cada PFA un valor real que cuantifique su calidad y, posteriormente, analizar los números resultantes para su comparación.

Antes de explicar los indicadores de calidad considerados, a continuación se presentan algunas definiciones que deben saberse.

Definición 1.2.7 Para dos vectores factibles $\mathbf{x}, \mathbf{x}' \in S$, se dice que $\mathbf{x}$ *domina débilmente* a $\mathbf{x}'$ y lo denotamos como $\mathbf{x} \succeq \mathbf{x}'$ si y solo si $f_i(\mathbf{x}) \geq f_i(\mathbf{x}')$ para todo $i = 1, \ldots, m$. Además, se dice que el conjunto A *domina débilmente* al conjunto B, $A \succeq B$, cuando cada punto $\mathbf{x}_2 \in B$ es débilmente dominado por al menos un punto $\mathbf{x}_1 \in A$.

Definición 1.2.8 Un vector de decisión $\mathbf{x} \in S$ se dice *débilmente eficiente* si y solo si no existe otro vector factible $\mathbf{x}' \in S$ tal que $\mathbf{x}'$ domine débilmente a $\mathbf{x}$.

Definición 1.2.9 Sea Ω el conjunto de todas las aproximaciones de frentes de Pareto, un *indicador de calidad unario* es una función $\mathscr{I} : \Omega \to \mathbb{R}$ que asigna a cada aproximación del frente de Pareto del conjunto $PFA \in \Omega$ un valor real $\mathscr{I}(PFA)$.

Una característica deseable de los indicadores de calidad es que deben asignar un valor indicador a $f(A)$ que sea mejor (o igual) que el asignado a $f(B)$ cuando el conjunto de aproximación

A del frente óptimo de Pareto sea mejor que el conjunto de aproximación B en términos de dominio débil, es decir, cuando B esté débilmente dominado por A. Estos indicadores se llaman *cumplidores de Pareto* y se definen de la siguiente manera.

Definición 1.2.10 Un indicador de calidad $\mathscr{I}$ se llama *cumplidor de Pareto* si y solo si $A \succeq B$ implica que $\mathscr{I}(\mathbf{f}(A)) \geq \mathscr{I}(\mathbf{f}(B))$ para cualquier conjunto de aproximación A, B.

Como los algoritmos estudiados son heurísticos, podrían devolver diferentes soluciones finales incluso cuando se da la misma configuración inicial. Por tanto, para garantizar que las conclusiones obtenidas son independientes de la ejecución, cada instancia ha sido ejecutada N_{runs} veces. Así, para cada algoritmo l, con $l \in \{1, \ldots, l_{mx}\}$, se han obtenido N_{runs} diferentes aproximaciones de conjuntos de Pareto $PS_l^1, \ldots, PS_l^{N_{\text{runs}}}$ en el espacio de decisión. Todos estos conjuntos resultantes para todos los algoritmos componen el conjunto de todas las aproximaciones del conjuntos de Pareto denotadas como *SPS*. Dado que el frente de Pareto real es necesario para el cálculo de algunos indicadores de calidad, en problemas como los nuestros en los que se desconoce, se utiliza un conjunto de referencia aproximado *RS*. En este trabajo, se ha generado *RS* fusionando todos los individuos de las aproximaciones del conjunto de Pareto *SPS*, seleccionando aquellos que no están dominados y obteniendo su imagen en el espacio objetivo.

Para evaluar la calidad global, hemos utilizado el conocido indicador del *hipervolumen* basado en el cálculo del hipervolumen de la parte del espacio de decisión que está débilmente dominado por la aproximación del frente de Pareto [54, 55].

1.2.3.1 Indicador de calidad de hipervolumen

Este indicador de calidad global mide el hipervolumen de la porción del espacio objetivo que está débilmente dominado por el conjunto de aproximación. En este sentido, cuanto mayor sea el hipervolumen, mejor será la aproximación.

Para medir esta cantidad, se necesita una aproximación del punto nadir, que está dominado por todos los puntos. Tiene que ser el mismo punto nadir aproximado para todas las ejecuciones y todas las configuraciones para permitir una comparación justa. En los estudios computacionales presentados, el punto cuyo componente i-ésimo es el mínimo de todos los componentes i-ésimo de los puntos en $f(SPS)$ se considera una aproximación del punto nadir obtenido al considerar todas las aproximaciones del frente de Pareto juntas.

En el algoritmo 1, se da una descripción de cómo calcular el hipervolumen cuando se consideran dos funciones objetivos, f_1 y f_2 para un problema de minimización. El primer paso es calcular este punto nadir aproximado $\mathbf{z}^{\text{nad}} = (f_1^{(\text{máx})}, f_2^{(\text{máx})})$, donde $f_i^{(\text{máx})}$ denota el valor mínimo de f_i con $i = 1,2$ cuando se consideran todas las soluciones en *SPS*. Entonces, el hipervolumen se calcula como el área cubierta por los puntos del frente de Pareto y el punto nadir considerado $\mathbf{z}^{\text{nad}}$. En la figura 1.9 se da una representación gráfica de esta métrica para un problema de optimización bi-objetivo. En particular, la figura de la izquierda ilustra el procedimiento de cálculo de la métrica del hipervolumen y la figura de la derecha muestra cómo el hipervolumen crece a medida que lo hace el número de puntos en el frente de Pareto.

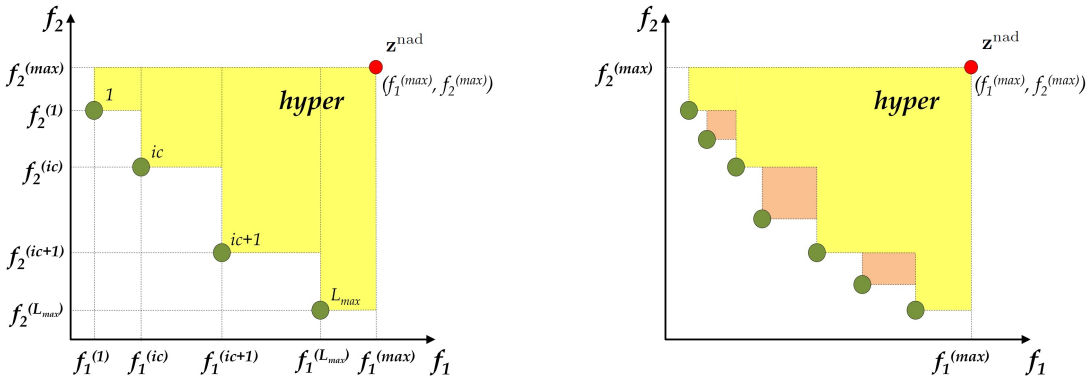


Figura 1.9: Cálculo del hipervolumen.

Algoritmo 1 Cálculo del indicador de hipervolumen

1: Calcular el punto nadir $\mathbf{z}^{\text{nad}} = (f_1^{(\text{máx})}, f_2^{(\text{máx})})$
2: Asignar $hyper = 0$
3: **for** $ic = 1$ hasta $(L_{\text{máx}} - 1)$ **do**
4: $\quad hyper = hyper + (f_2^{(\text{máx})} - f_2^{(ic)}) \cdot (f_1^{(ic+1)} - f_1^{(ic)})$
5: **salida** $hyper = hyper + (f_2^{(\text{máx})} - f_2^{(L_{\text{máx}})}) \cdot (f_1^{(\text{máx})} - f_1^{(L_{\text{máx}})})$

1.3 Algoritmos de búsqueda

Los algoritmos de búsqueda son una gran familia de mecanismos generales que, en la mayoría de los casos, pueden aplicarse a problemas de optimización global monoobjetivo y multiobjetivo [56]. Surgen como una alternativa a las metodologías locales clásicas.

El principal inconveniente de las metodologías locales clásicas es que fallan al encontrar la mejor solución global de un problema pues se quedan atrapados fácilmente en óptimos locales. Además, no pueden utilizar la información global del problema necesaria para encontrar el óptimo global en una función con múltiples óptimos locales. Uno de los algoritmos más ampliamente usado en optimización local es el método de descenso por gradiente. Se basa en el hecho de que el gradiente de una función se corresponde con la dirección en la cual la función tiene la mayor pendiente hacia el óptimo por lo que es fácil llegar a él tras unas iteraciones. Sin embargo, por esa misma característica, salir de dicho óptimo (local) es imposible para este método. Es por ello que surgen como alternativa los métodos de búsqueda global que sí permiten encontrar el óptimo global entre todos los óptimos locales.

Los métodos de optimización global se pueden clasificar en base a distintos criterios. En la bibliografía se pueden encontrar propuestas de distintas taxonomías, cada una con su punto de vista. No obstante, parece claro que existe una primera clasificación que divide los métodos en dos grandes grupos: deterministas y heurísticos.

En los métodos deterministas tales como la optimización de Lipschitzian o Ramificación y Poda (B&B, *Branch and Bound*), apenas se incluyen factores aleatorios. Algunos de ellos convergen en el óptimo global bajo ciertas condiciones, pero cuando el algoritmo se detiene después de un número finito de iteraciones, la precisión de la solución puede que no se conozca

con exactitud. En consecuencia, necesitan de información global del problema. En relación a esto, los métodos completos más eficientes normalmente combinan técnicas de ramificación con una o varias técnicas de optimización local, análisis convexo, análisis de intervalos o programación de restricciones.

Además, los algoritmos deterministas garantizan encontrar para cada instancia de tamaño finito de un problema una solución óptima en un tiempo limitado. Sin embargo, para los problemas NP-completos, no existe ningún algoritmo de tiempo polinómico. Por lo tanto, los métodos deterministas necesitan tiempos demasiado altos para dar solución a problemas prácticos. Así, el uso de métodos heurísticos para resolver los problemas ha ido aumentando cada vez más en los últimos años. En este tipo de métodos se sacrifica la garantía de encontrar la solución óptima a cambio de obtener una buena solución en un tiempo significativamente menor.

Entre las metodologías heurísticas, cabe mencionar dos tipos de algoritmos:

- *Heurísticas constructivas*: Parten de una solución inicial vacía y, durante el procedimiento de optimización añaden algunos componentes para construir la solución final. Un ejemplo de un método heurístico constructivo es el algoritmo *Greedy* [57].
- *Métodos de búsqueda local*: Parten de una solución inicial no vacía y, en cada iteración tratan de reemplazarla con otra solución de su vecindario que tenga más calidad. Por ejemplo, la Búsqueda Estocástica de Agente Único (SASS, *Single Agent Stochastic Search*) [58] y su versión multiobjetivo MOSASS [59].

Además de ambos enfoques, y a menudo utilizando sus mecanismos, otra subfamilia de algoritmos heurísticos que se puede distinguir es el metaheurístico. Debido a su relevancia y al hecho de que los algoritmos utilizados en esta tesis pertenecen a este grupo, la siguiente subsección se dedicará a ellos. Para ello, se va a seguir la toxonomía propuesta en [60] y que se encuentra representada en la figura 1.10. En las siguientes secciones se describirá la rama de los algoritmos heurísticos y más concretamente el camino descendente hacia los algoritmos meméticos.

1.3.1 Algoritmos Heurísticos: computación evolutiva

En informática, hay dos objetivos fundamentales que los algoritmos deben alcanzar: (i) obtener una calidad de solución buena u óptima y que se pueda demostrar y (ii) obtener tiempos de ejecución factibles. Una heurística es un algoritmo que abandona uno o ambos objetivos. Por ejemplo, una heurística generalmente encuentra soluciones bastante buenas, pero no hay pruebas de que las soluciones no puedan llegar a ser arbitrariamente malas; o generalmente se ejecuta razonablemente rápido, pero no hay argumento de que este siempre será el caso.

Como se mencionó anteriormente, puede suceder que no haya interés en alcanzar la solución óptima porque el tamaño del problema a resolver está más allá del límite computacional de los algoritmos de optimización conocidos dentro del tiempo disponible del computador. Además, en otros casos, el problema podría resolverse de manera óptima, pero se puede considerar que el esfuerzo (tiempo, dinero, etc.) requerido para encontrar la solución óptima no vale la pena. En tales casos, se puede utilizar un algoritmo heurístico, que posiblemente encuentre una solución factible cercana al óptimo en términos de valor de la función objetivo. De hecho, a menudo

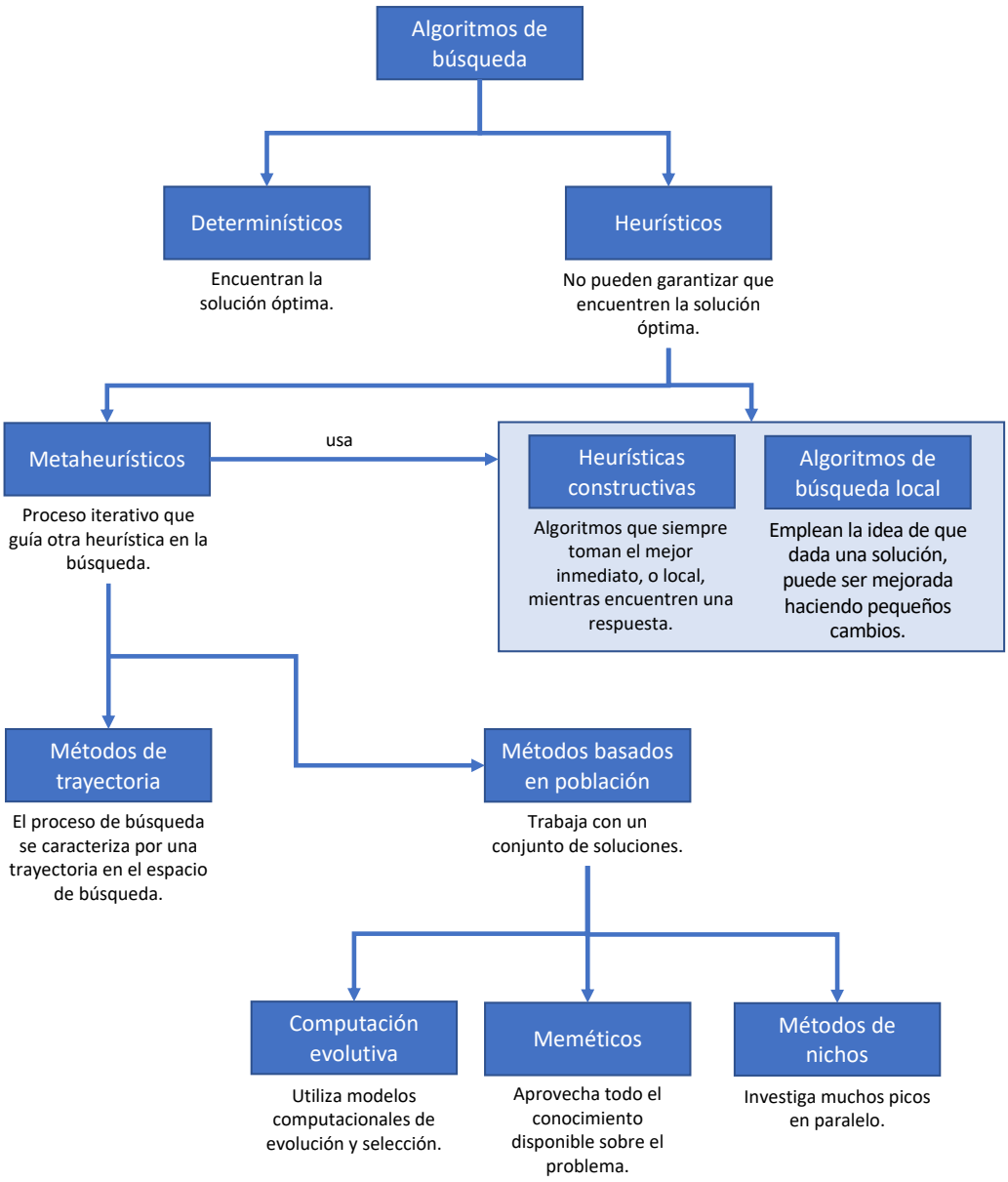


Figura 1.10: Una taxonomía de los algoritmos evolutivos.

ocurre que un algoritmo heurístico bien diseñado puede dar resultados de buena calidad (casi óptimos).

Las principales ventajas de los algoritmos heurísticos son que dichos algoritmos son (a menudo) conceptualmente más simples y (casi siempre) mucho más rápidos computacionalmente que los algoritmos exactos. Entre los métodos heurísticos, es posible distinguir entre heurística constructiva, métodos de búsqueda local y metaheurística.

- Las heurísticas constructivas son algoritmos que siempre toman la mejor solución inmediata o local mientras encuentran una respuesta. Generan soluciones desde cero añadiendo oportunamente componentes de soluciones definidas a una solución parcial inicialmente vacía. Esto se hace hasta que se completa una solución o se satisfacen otros criterios de detención [61].
- Un método de búsqueda local emplea la idea de que una solución global dada puede

mejorarse haciendo pequeños cambios. Las soluciones obtenidas al modificar una solución s, se llaman vecinas de s, y la aplicación de un operador que produce una vecina s' se denomina comúnmente movimiento. Un algoritmo de búsqueda local comienza con una solución inicial y se mueve de vecino en vecino el mayor tiempo posible mientras mejora el valor de la función objetivo. El principal inconveniente de esta estrategia es su dificultad para escapar de los óptimos locales donde la búsqueda no puede encontrar ninguna solución vecina adicional que mejore el valor de la función objetivo.

- Una metaheurística se puede definir como un marco o enfoque algorítmico de alto nivel. Consiste en un proceso iterativo, que guía otras heurísticas en la búsqueda de soluciones factibles. Las metaheurísticas son aproximadas y, por lo general, no deterministas, y normalmente se aplican a problemas para los cuales no existe un algoritmo específico o heurístico satisfactorio; o cuando no es práctico implementar dicho método. No son específicos del problema, pero pueden hacer uso del conocimiento específico del dominio en forma de heurística controlada por una estrategia de nivel superior.

1.3.2 Algoritmos metaheurísticos

El término metaheurístico, introducido por primera vez en [62], tiene su origen en la composición de dos palabras griegas. Heurístico proviene del verbo *heuriskein* que significa "encontrar", mientras que el prefijo *meta* significa "más allá, en el nivel superior". Algunos algoritmos muy conocidos son considerados metaheurísticos: optimización aleatoria, algoritmos genéticos, enfriamiento simulado, búsqueda Tabú, optimización de colonia de hormigas, GRASP, búsqueda de difusión estocástica, optimización extrema generalizada y búsqueda de Harmony. Se han propuesto innumerables variantes e híbridos de estas técnicas, y se han publicado muchas más aplicaciones de metaheurísticas para problemas específicos. Este es un campo activo de la investigación, con una considerable bibliografía, una gran comunidad de investigadores y usuarios, y un amplio rango de aplicaciones.

Una metaheurística exitosa tiene que proporcionar un equilibrio entre la explotación y la exploración del espacio de búsqueda para identificar regiones con mayor nivel de calidad de las soluciones. El equilibrio entre exploración y explotación es muy importante, por un lado para identificar regiones rápidamente en el espacio de búsqueda con soluciones de gran calidad y por otro lado para no perder demasiado tiempo en las regiones de búsqueda del espacio que ya han sido exploradas y no proporcionan soluciones de gran calidad [63]. La principal diferencia entre las metaheurísticas existentes se basa en la forma particular en que tratan de lograr este equilibrio. Los diferentes enfoques metaheurísticos pueden caracterizarse por diferentes aspectos basados en el camino de búsqueda que siguen, cómo se explota la memoria, el uso de poblaciones de soluciones, el número de vecinos considerados y, por qué no, las fuentes inspiradas. En [64] se presenta una discusión sobre estos aspectos.

1.3.3 Métodos basados en poblaciones

En secciones anteriores, se ha hablado brevemente sobre métodos de resolución de problemas clásicos, incluyendo métodos exactos y algoritmos de búsqueda local. Para estos métodos, dado un espacio de búsqueda y una función objetivo, algunos de ellos siempre devolverían la misma solución (por ejemplo, métodos determinísticos), mientras que otros podrían generar diferentes

soluciones basadas en la configuración inicial o punto de partida.

Hay una interesante observación compartida por todas estas técnicas: cada una se basa en una única solución como base para la exploración futura con cada iteración. O bien procesan soluciones completas en su totalidad (búsqueda local), o bien construyen la solución final a partir de bloques de construcción más pequeños (B&B). A pesar de sus diferencias, cada uno de estos algoritmos trabaja o construye una sola solución a la vez.

Las Metaheurísticas Basadas en Poblaciones (PBM, *Population Based Metaheuristic*) [63] son algoritmos que trabajan con un conjunto de soluciones (es decir, una población) al mismo tiempo en vez de con una única solución. A primera vista, quizás parezca que esta idea no proporciona realmente nada nuevo, ya que los algoritmos anteriores se podrían ejecutar k veces para incrementar la probabilidad de encontrar el óptimo global. Pero hay un componente adicional que puede hacer que los algoritmos basados en poblaciones sean esencialmente diferentes de otros métodos de resolución: el concepto de competencia entre soluciones de una población. Es decir, simular el proceso evolutivo de la competencia y selección y dejar que las soluciones candidatas en la población luchen por tener espacio en las generaciones futuras. De esta manera, los algoritmos basados en poblaciones proporcionan una forma natural e intrínseca de explorar el espacio de búsqueda.

Algunos de los métodos basados en poblaciones más estudiados son la computación evolutiva y la optimización de colonias. El primero se estudiará a continuación. El último está fuera del enfoque de este trabajo, pero se recomienda la lectura de los trabajos [61, 65-67] para una descripción en profundidad del algoritmo y sus aplicaciones.

1.3.4 Computación evolutiva

La Computación Evolutiva (EC, *Evolutionary Computation*) es una técnica moderna de búsqueda que usa modelos computacionales de procesos de evolución y selección [68, 69]. Los conceptos y mecanismos de la evolución darwiniana y la selección natural están codificados en Algoritmos Evolutivos (EA, *Evolutionary Algorithms*) y se utilizan para resolver problemas en muchos campos de la ingeniería y la ciencia [70].

El fuerte parecido a los procesos biológicos, así como sus aplicaciones iniciales para el modelado de sistemas adaptativos complejos influyó en la terminología utilizada por los investigadores de la EC. Tiene mucho de genética, de teoría evolutiva y de biología celular, tanto es así que una solución candidata a un problema se llama individuo, mientras que el conjunto entero (superconjunto) de soluciones actuales se llama población. Para algunos dominios de problemas, una población puede dividirse en varias subpoblaciones o especies. La representación real (codificación) de un individuo se llama genoma o cromosoma. Cada genoma consiste en una secuencia de genes, es decir, atributos que describen a un individuo. Un valor de un gen se llama alelo. Cuando las soluciones individuales son modificadas para producir nuevas soluciones candidatas, se dice que se reproducen y la nueva solución candidata se denomina descendencia o hijo. Durante la evaluación de una solución candidata, esta recibe una calificación denominada aptitud, que indica la calidad de la solución en el contexto del problema dado. Cuando la población actual es reemplazada por sus descendientes, la nueva población se llama nueva generación. Finalmente, todo el proceso de búsqueda de una solución óptima se denomina evolución [71].

Algoritmo 2 Algoritmo evolutivo

1: Generar una población inicial
2: Evaluar la población
3: **while** la condición de parada no se cumpla **do**
4: Recombinar la población para obtener una nueva generación
5: Mutar la generación para obtener una nueva población
6: Evaluar la nueva población
7: Seleccionar los individuos de la nueva población que serán considerados para la siguiente generación
8: **salida:** mejor solución encontrada

El Algoritmo 2 describe la estructura básica de un EA. Inicialmente, se crea una población de individuos generados aleatoriamente (o individuos obtenidos de otras formas tales como heurísticas constructivas). La aptitud se usa para determinar el mérito relativo de cada individuo. Una vez que se ha obtenido la población inicial, se lleva a cabo un proceso iterativo. En cada iteración, se genera una nueva descendencia usando un operador de recombinación, normalmente un cruce entre dos o varios padres [72, 73]. Otras técnicas usan estadísticas de población para generar hijos [74, 75].

Con el fin de evitar la convergencia a un óptimo local, se suele introducir algo de ruido en el proceso de búsqueda, que en este tipo de algoritmos consiste en aplicar un operador de mutación a los individuos progenitores. En algunas aplicaciones, se utilizan pequeños cambios aleatorios como mecanismo de mutación, pero en otros, resulta ser bastante beneficioso usar métodos de mejora para aumentar la aptitud de los individuos. Los algoritmos evolutivos que aplican un algoritmo de búsqueda local para cada individuo de una población se denominan algoritmos meméticos. Estos se basan en la idea de que mientras que el uso de una población asegura una exploración del espacio de búsqueda, el uso de técnicas de búsqueda local ayuda a identificar rápidamente áreas buenas en el espacio de búsqueda. Sin embargo, cuando se aplica búsqueda local puede ocurrir una convergencia prematura hacia soluciones subóptimas. Con el fin de evitar este inconveniente, existen, además del uso de un operador de mutación aleatoria, numerosas maneras de mantener la diversidad de la población. Algunas de estas formas son la agrupación [76] o el uso de nichos [77]. Finalmente, los individuos compiten cada uno entre el resto y también contra los padres para mantenerse en la población en la siguiente iteración. Esto es hecho mediante un esquema de selección.

Existen técnicas similares que difieren en detalles de implementación y en la naturaleza del problema, como la programación genética [78, 79], la programación evolutiva [80], la estrategia evolutiva [81] y el sistema clasificador de aprendizaje [82]. Sin embargo, existe un tipo de algoritmos que incorpora conocimientos de varias de estas técnicas. Estos son los algoritmos meméticos, que se describirán brevemente en la siguiente sección.

1.3.5 Algoritmos Meméticos

Bajo la terminología de Algoritmo Memético (MA, *Memetic Algorithm*) se engloba una amplia clase de metaheurísticas (ver Sección 1.3.2). Los MA se basan en una población de agentes, en la que un conjunto de agentes cooperantes y competidores dedicaron periodos de

tiempo a la mejora individual de las soluciones mientras interactuaban esporádicamente [83].

A diferencia de los métodos tradicionales de Computación Evolutiva (EC, *Evolutionary Computation*), los MA están intrínsecamente interesados en explotar todo el conocimiento disponible sobre el problema en cuestión como una forma de acelerar el proceso de búsqueda [84]. Los MA explotan el conocimiento del problema incorporando heurísticas preexistentes, reglas de reducción de datos de preprocesamiento, algoritmos manejables de aproximación y parámetros fijos, técnicas de búsqueda local, operadores de recombinación especializados, métodos exactos truncados, etc. Además, un factor importante es el uso de la representación adecuada del problema que se esté abordando. El término *hibridación* se usa comúnmente para denotar el proceso de incorporación de conocimiento del problema al algoritmo [85]. Por esta razón, los MA a veces se denominan *Algoritmos evolutivos híbridos*. La incorporación del conocimiento del dominio del problema no es un mecanismo opcional, sino una característica fundamental que caracteriza a los MA. Las ventajas de este enfoque se descuidaron notablemente en los EA durante mucho tiempo a pesar de algunas voces contrarias [86]. La formulación del llamado *No-Free-Lunch Theorem* [87] dejó en claro que un algoritmo de búsqueda funciona estrictamente de acuerdo con la cantidad y la calidad del conocimiento del problema que incorporan, respaldando así uno de los *leivmotivs* de los MA.

La filosofía funcional está perfectamente ilustrada por el término "memético". Acuñado por Dawkins [88], la palabra *meme* denota una analogía con el gen en el contexto de la evolución cultural [89]. En palabras de Dawkins:

"Ejemplos de memes son melodías, ideas, frases, modas de ropa, formas de hacer macetas o construir arcos. Del mismo modo que los genes se propagan en el conjunto de genes al saltar de un cuerpo a otro a través de los espermatozoides u óvulos, los memes se propagan a sí mismos en el conjunto de memes al saltar de un cerebro a otro a través de un proceso que, en sentido amplio, se puede llamar imitación"

1.3.5.1 Un algoritmo memético básico

Como se mencionó, los MA combinan diferentes estrategias de búsqueda en un enfoque algorítmico combinado [84]. Al igual que los EA, los MA son algoritmos metaheurísticos basados en poblaciones, lo que significa que mantienen una población de soluciones para el problema en cuestión. Dado que es posible tener soluciones factibles o proto-soluciones (estructuras que se pueden extender/modificar para producir soluciones viables) o incluso soluciones inviables (que se pueden "reparar" para restaurar la viabilidad), el término "soluciones" se adopta aquí [90]. También se supone que tanto los procesos de reparación como de extensión se pueden realizar con bastante rapidez, para justificar su inclusión en la población. Cada una de estas soluciones se denominará individuo como la jerga de EA, principalmente para simplificar la discusión. En el contexto de los MA, un *agente* representa una unidad de procesamiento que puede contener múltiples soluciones y tiene métodos de dominio de problemas que ayudan a mejorarlos si es necesario [83]. Cada individuo o agente representa una solución/método tentativo para el problema en consideración.

Un algoritmo memético comienza con un proceso de inicialización [84]. A diferencia de los EA estándar, que simplemente generarían una serie de soluciones aleatorias, en los MA se

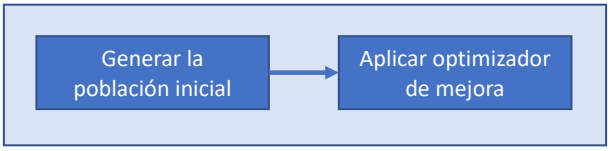


Figura 1.11: Inicialización de un proceso de un MA.

usan típicamente mecanismos más sofisticados, por ejemplo, heurística constructiva [91, 92] (ver figura 1.11). Después de eso, se lleva a cabo un proceso iterativo hasta que se cumple un criterio de terminación (ver figura 1.12).

1.3.5.2 Proceso de reinicio de la población

La inclusión de varios componentes que incorporan conocimiento contribuyen a acelerar la convergencia de la población. Por lo tanto, este bloque es fundamental en los MA. Entonces, si la población converge, es mejor actualizarla en lugar de mantenerla restringida a una pequeña región en el dominio de búsqueda. Este proceso de reinicio se puede realizar de diferentes maneras. Uno de ellos podría ser: se mantiene una cierta fracción p de la población (este valor no debería ser muy alto ya que de lo contrario la población obviamente convergería nuevamente), y las soluciones restantes se crean desde cero, como ocurre en la fase de inicialización (ver figura 1.11).

1.3.5.3 Proceso de paso generacional

Esta es la parte del algoritmo en el que tiene lugar la evolución de las soluciones. Como se puede ver en la figura 1.12, este bloque generacional incluye tres pasos: selección, reproducción y actualización. El primero y el tercero son responsables de los aspectos de competencia de los individuos en la población.

El operador *selección* evalúa la bondad de los individuos en la población, utilizando la información proporcionada por una función de guía dependiente del problema (función de aptitud en la terminología EA), y determina la muestra de individuos que se reproducirán. El componente *actualización* mantiene a la población en un tamaño constante mediante la sustitución de algunos individuos preexistentes en la población original por algunos de los obtenidos de la nueva población (utilizando algún criterio específico). En la etapa *reproducción*, se crean nuevos individuos (o agentes) utilizando algunos operadores reproductivos (mutación, búsqueda local, cruce de dos puntos, etc.).

Para obtener más información sobre el diseño de los MA, se recomienda consultar los trabajos [84, 90, 93, 94]).

1.4 Computación de Alto Rendimiento

La Computación de Alto Rendimiento (HPC, *High Performance Computing*) surge como un conjunto de sistemas, herramientas y estrategias dirigidas a descomponer problemas computacio-

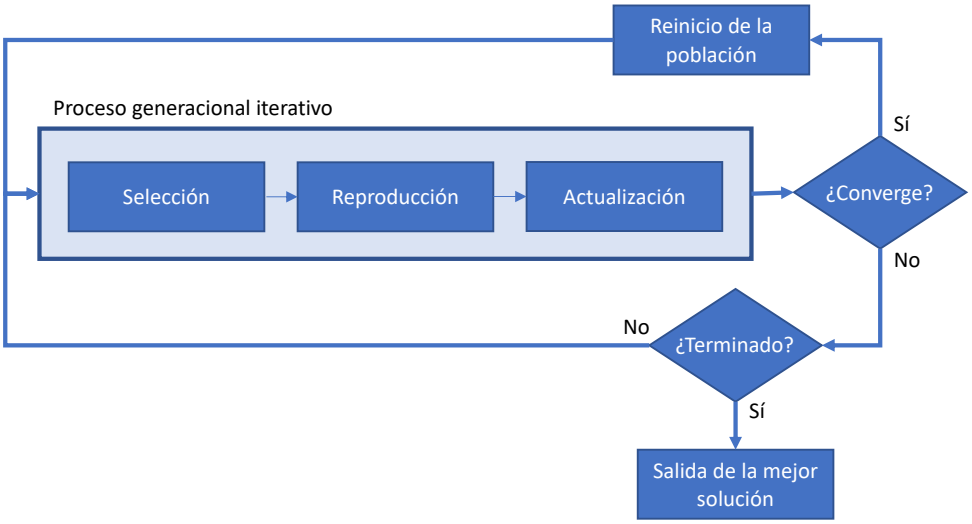


Figura 1.12: Estructura general de un MA.

nalmente costosos en varias tareas más pequeñas que podrían procesarse simultáneamente (es decir, en paralelo) aprovechando los marcos computacionales modernos.

Esta área de investigación, también conocida como cómputo paralelo, cubre el estudio de los niveles de hardware y software de los sistemas informáticos. Por un lado, con respecto al hardware, el HPC trata con arquitecturas paralelas. Por otro lado, a nivel de software, se refiere a los modelos de programación paralela.

En esta sección, se proporciona una revisión del estado del arte en HPC. En primer lugar, en la subsección 1.4.1, se presentan las arquitecturas paralelas existentes. En segundo lugar, en la subsección 1.4.2, se distinguen los dos paradigmas de programación paralela bien conocidos, a saber, la memoria compartida y la programación distribuida.

1.4.1 Arquitecturas paralelas

Los objetivos principales de la computación paralela son lograr aplicaciones eficientes al reducir el tiempo de computación y resolver problemas muy costosos que no pueden resolverse en un procesador secuencial. Gracias a los avances en HPC, podemos hacer frente a problemas complejos que implican un gran esfuerzo computacional y largos tiempos de ejecución. Sin embargo, para obtener un buen rendimiento de los recursos de HPC, se requiere cierto conocimiento sobre el hardware y su arquitectura paralela. Antes de programar un código paralelo, es recomendable comprender las características de las diferentes arquitecturas para seleccionar la más adecuada según la aplicación y lograr el máximo beneficio del sistema paralelo.

En la literatura de HPC, la clasificación más extendida para computadores paralelos es la llamada taxonomía de Flynn, que fue propuesta en 1972 [95]. En la figura 1.13, se representa considerando la extensión propuesta en [61], que incluye algunas subcategorías derivadas de la evolución de la computación paralela. La taxonomía de Flynn distingue cuatro arquitecturas dependiendo del número de instrucciones y flujos de datos:

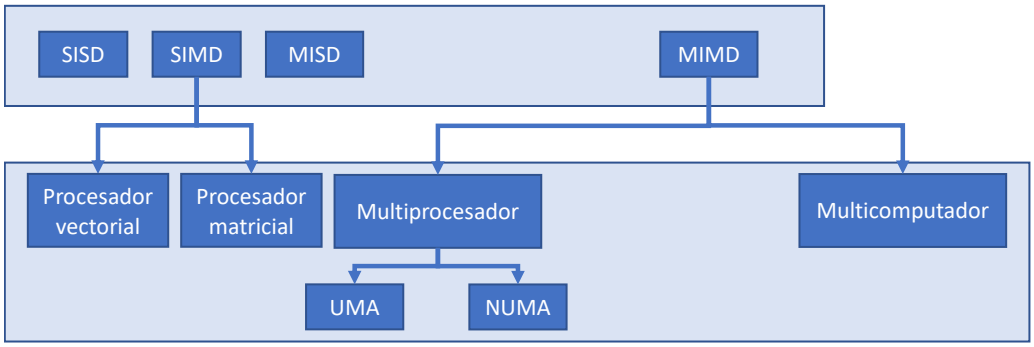


Figura 1.13: Taxonomía de Flynn considerando algunas extensiones.

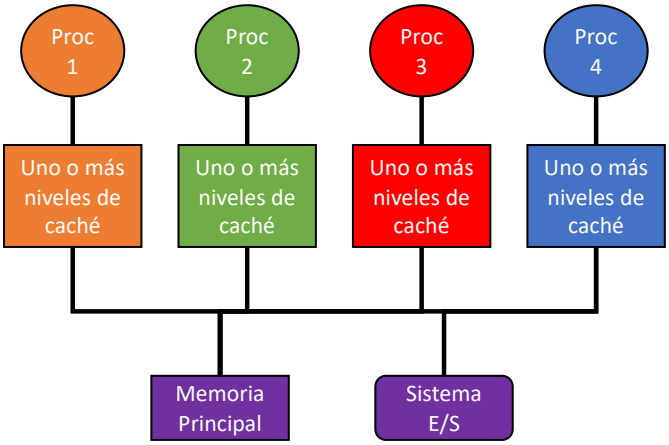
- SISD (*Single Instruction, Single Data stream*). En este modelo, los datos están disponibles en una única memoria y, además, se accede a ellos mediante una única secuencia de instrucciones ejecutadas en un único procesador. Es la arquitectura más básica y podemos encontrarla en computadores que tienen una sola CPU (es decir, sistemas uniprocesadores).
- MISD (*Multiple Instruction, Single Data stream*). En esta arquitectura se pueden ejecutar varias instrucciones de manera simultánea sobre el mismo bloque de datos. Consiste en un paralelismo intrínseco dado por el propio procesador a nivel de arquitectura.
- SIMD (*Single Instruction, Multiple Data stream*). En este grupo, las arquitecturas se caracterizan por un único flujo de instrucciones que opera en vectores de datos utilizando todos los procesadores. Más precisamente, todos ellos ejecutan la misma instrucción en cada paso de computación, es decir, las unidades de ejecución paralelas se sincronizan [96].
- MIMD (*Multiple Instruction, Multiple Data stream*). En estas arquitecturas, se pueden ejecutar diferentes programas con diferentes datos utilizando diferentes procesadores. Además, esos procesadores funcionan de forma asíncrona e independiente.

Si bien a principios del siglo XXI las arquitecturas SIMD y MIMD eran bastante populares, esto ha ido cambiando y ahora prácticamente cualquier unidad de procesamiento, ya sea personal o diseñada especialmente para el HPC disfruta de arquitectura MIMD. En consecuencia, a continuación describiremos brevemente esta arquitectura.

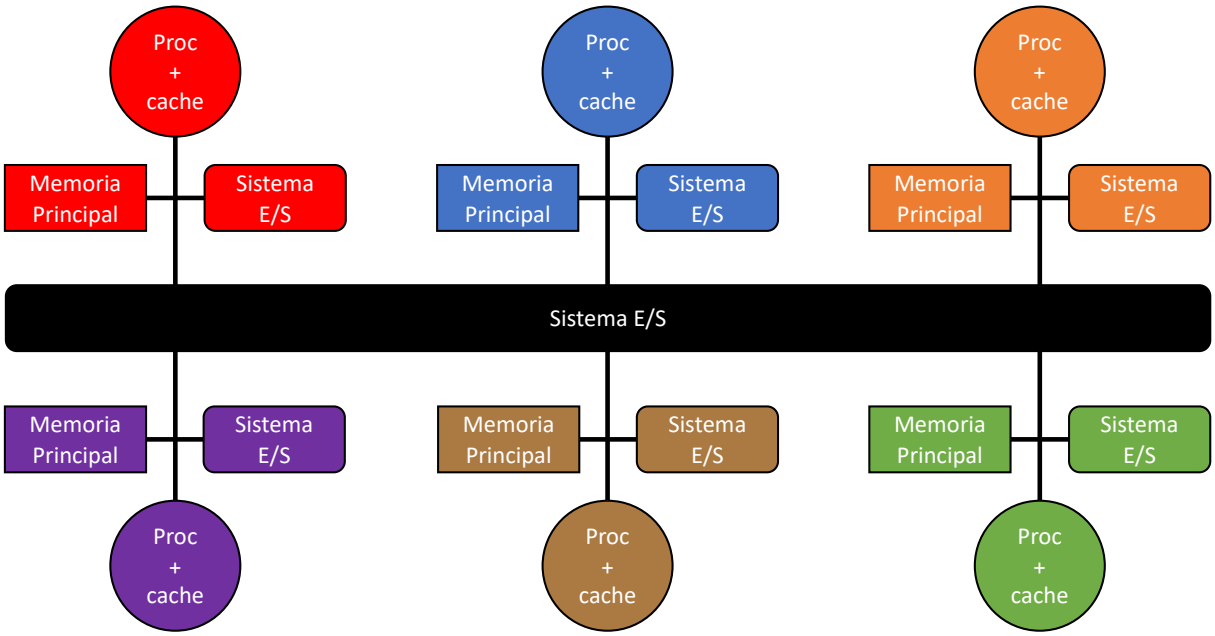
1.4.1.1 Arquitecturas MIMD

Las arquitecturas MIMD comprenden dos categorías: *multiprocesadores* y *multicomputadores* (ver figura 1.13). Por un lado, la característica principal de la arquitectura multiprocesador es que todos los procesadores tienen acceso directo a toda la memoria. Más precisamente, todos los bancos de memoria son comunes y accesibles para todos los procesadores por medio de una red de interconexión. Por otro lado, los multicomputadores abarcan aquellas arquitecturas en las que los diferentes módulos de memoria son propiedad local de un procesador diferente. Luego, cuando un procesador quiere acceder a un banco de memoria remoto, no solo tiene que usar una red de interconexión, sino que también se requiere un método de paso de mensajes (consulte la Sección 1.4.2.2).

Los multiprocesadores se pueden dividir en dos grupos: *memoria compartida centralizada* y *memoria compartida distribuida* [97]. En los multiprocesadores centralizados de memoria compartida, varios subsistemas compuestos por un procesador y su caché están conectados a la



(a) Acceso uniforme a la memoria (UMA).



(b) Arquitectura de acceso a memoria no uniforme (NUMA).

Figura 1.14: Organización de los dos tipos de arquitecturas multiprocesadores.

misma memoria física, generalmente a través de buses o un conmutador como se puede ver en la figura 1.14a. Su característica principal es que el tiempo de acceso a toda la memoria es uniforme sin importar el procesador de origen. Por lo tanto, con frecuencia se conocen como *Acceso Uniforme a Memoria (UMA, Uniform Memory Access)*. Frente a ellos, los multiprocesadores de memoria compartida distribuida están compuestos por nodos individuales, donde hay un procesador, una memoria local y, por lo general, algún sistema de entrada/salida. Todos esos nodos están vinculados por medio de una red de interconexión (ver figura 1.14b). Según esta distribución, un procesador puede acceder a su propia memoria local más rápido que la memoria no local. Esto significa que el tiempo de acceso a la memoria no es uniforme ya que depende de la posición relativa entre la memoria y el procesador. Estas arquitecturas se denominan *Acceso a Memoria No Uniforme (NUMA, Non-Uniform Memory Access)*.

1.4.2 Modelos y herramientas de programación paralela.

Diseñar un programa paralelo es inherentemente más difícil que la programación convencional, ya que generalmente requiere una comprensión profunda de sus procedimientos y comportamiento. El último, al que nos hemos referido informalmente como convencional, se conoce en realidad como programación secuencial. Su característica principal es que mantiene un proceso único, lo que significa un flujo de control único. Por el contrario, la programación paralela utiliza implementaciones concurrentes, donde dos o más procesos trabajan juntos para realizar una sola tarea. Luego, la comunicación y la sincronización entre las diferentes subtareas se necesitan con frecuencia. Esos aspectos son a menudo los mayores desafíos para lograr un buen desempeño paralelo.

Dependiendo de la arquitectura, surgen tres modelos de programación paralela. Dos modelos independientes como son la programación de memoria compartida para multiprocesadores y la programación de memoria distribuida para sistemas distribuidos, que se explicarán brevemente en las siguientes subsecciones, y un modelo híbrido combinación de los dos anteriores.

1.4.2.1 Programación de memoria compartida

En la programación de memoria compartida, todos los procesos pueden acceder directamente a toda la memoria, permitiendo la comunicación entre ellos. Hay que tener en cuenta que estos programas pueden ejecutarse en el mismo procesador físico o en otros.

El paradigma de programación más extendido para implementar la concurrencia de aplicaciones es el Multihilo (MT, *MultiThreadings*). En otras palabras, el MT es un mecanismo para explotar el paralelismo en multiprocesadores de memoria compartida. Un programa tradicional de subproceso único se puede definir como un flujo de control independiente asociado uno a uno con un contador de programa, una pila para realizar un seguimiento de las variables locales, un espacio de direcciones y un conjunto de recursos. De lo contrario, la programación MT permite que el programa principal ejecute múltiples tareas simultáneamente, dividiéndolo en múltiples subprocesos. Es decir, diferentes flujos de control que pueden ejecutar sus instrucciones de forma independiente y concurrente. Esto implica que se permita la superposición de las operaciones de entrada, salida y computación. Además, cuando un programa MT se ejecuta en un dispositivo multiprocesador, se pueden ejecutar varios subprocesos simultáneamente (en paralelo) en procesadores separados, explotando el paralelismo del hardware. Algunas de las herramientas existentes para implementar paralelismo en un marco de memoria compartida son los Pthreads [98], Hilos de Java u Open Multi-Processing (OpenMP) [99].

1.4.2.2 Programación distribuida

La programación distribuida se refiere principalmente a los métodos de paso de mensajes. Existen varias estrategias para comunicar mensajes entre diferentes elementos de procesamiento, como por ejemplo, el uso de sistemas informáticos de Internet, sistemas basados en objetos y bibliotecas de paso de mensajes. Algunos de los mecanismos más populares de programación distribuida son Parallel Virtual Machine (PVM), Message-Passing Interface (MPI), Java Remote Method Invocation (RMI) y Globo [100].

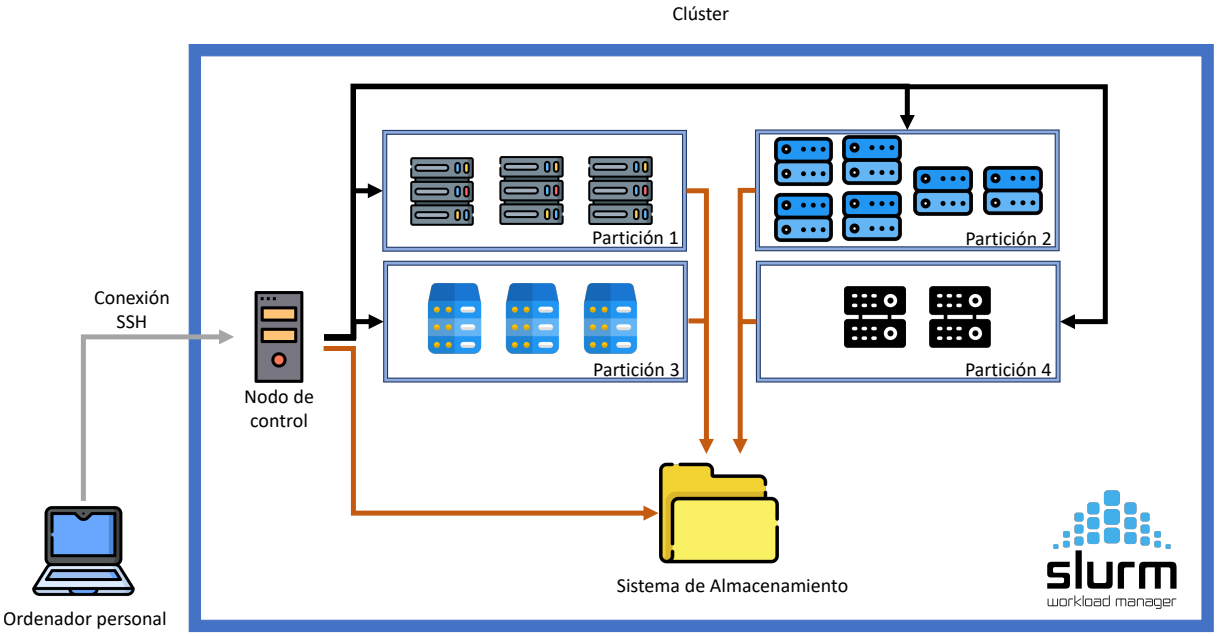


Figura 1.15: Sistema de Slurm.

1.4.3 Arquitecturas Paralelas utilizadas

Dejando atrás los ordenadores personales, actualmente es relativamente sencillo tener acceso a centros de supercomputación en el que la cantidad de recursos disponibles es considerablemente mayor a un computador clásico. Dadas las ventajas de estos centros, suelen estar altamente demandados y es por eso que requieren de un software que gestione todas las peticiones que deben de ejecutarse en dichos equipos. El software Slurm [101] surgió como solución a dicho problema y actualmente se encuentra instalado en la mayoría de centros de computación. Este software pone a disposición del usuario todos los recursos hardware disponibles para que se envíe toda la experimentación a través de scripts mediante un sistema de colas. Cada script contiene la información necesaria para realizar una ejecución secuencial de un problema utilizando los recursos que se soliciten en el mismo. Esto, por ejemplo, permite dividir un trabajo secuencial en dos paralelos gestionando su memoria, entradas, salidas y errores de forma completamente independiente. Un ejemplo del sistema Slurm se puede ver en la figura 1.15. En ella se puede observar un sistema en el que hay un nodo controlador al que accede un usuario y el resto de nodos están disponibles para realizar las ejecuciones. Cada nodo tiene sus propias especificaciones pero todos comparten el mismo almacenamiento.

En esta tesis, dada la alta carga computacional de las experimentaciones, se ha hecho uso de diferentes centros de supercomputación. A continuación se describe brevemente cada uno de ellos:

- NLHPC. Laboratorio Nacional de Computacion de Alto Rendimiento de la Universidad de Chile. Tiene una capacidad de 5236 CPU cores y 20480 GPUs cores alcanzando los 266 Teraflops. Además posee un almacenamiento de 274 TB y 23 TB de memoria RAM [102]
- SCBI. Supercomputing and Bioinnovation Center de la Universidad de Málaga. Tiene una capacidad de cómputo de 63 TFlops [103].
- Stallo. The Norwegian e-infrastructure for Research & Education. 312 TFlops con un total de 11424 cores, 26.2 TB de memoria RAM y 2000 TB de almacenamiento [104].

- Eagle. Poznan Supercomputing and Networking Center. 1087 nodos que de forma conjunta alcanzan 1.4 PFlops entre sus 30656 cores. Además los nodos tienen una memoria RAM desde 64GB hasta 256GB [105].
- Bullxual. Grupo de Supercomputación y Algoritmos de la Universidad de Almería. 484 CPU cores con 3.9 TB de memoria RAM y 22.9 TB de almacenamiento [106].

No obstante, estos sistemas tampoco están exentos de problemas. A continuación se muestra un listado de los problemas encontrados en el desarrollo de esta tesis.

- Bases de datos. Como se mencionó en la sección 1.1.3, las bases de datos son modificadas para adaptarlas a las necesidades de los problemas. En consecuencia, cualquier modificación de una base de datos en un centro de computación debe replicarse en el resto pues la comparación de resultados de algoritmos ejecutados en distintos centros deben de realizarse bajo las mismas condiciones.
- Tiempo de ejecución. Dado que los centros de computación son un conjunto de equipos con distintas especificaciones, aprovechar todos los recursos para obtener resultados rápidamente es una idea acertada siempre que no sea necesaria una estimación de tiempo pues las ejecuciones dependerán de las especificaciones de los nodos donde que se hayan utilizado.
- Espacio. En los equipos personales se puede gestionar los recursos de almacenamiento de forma sencilla pues solo un usuario tiene acceso a ellos. En cambio, en los centros de computación, hay que gestionar el espacio disponible y periódicamente realizar una limpieza y copia de seguridad pues en las actualizaciones de hardware puede perderse toda la información.
- Software disponible. Debido a los distintos equipos que se pueden encontrar en los centros de computación, en ocasiones es necesario instalar software que todavía no se encontraba disponible. Además, al utilizar software de terceros, en ocasiones este es totalmente incompatible con la arquitectura de los nodos de cómputo lo cual los hace completamente inoperables. Esto ha impedido utilizar algunos centros de computación muy potentes.
- Ejecuciones erróneas. En ocasiones, los resultados suelen ser incorrectos. Esto puede deberse desde errores no depurados en el propio software hasta fallos del sistema de archivo al guardar los resultados lo que obliga a repetir todos los experimentos.

1.4.4 Técnicas de balanceo de carga

El paralelismo tradicional que se ha descrito brevemente en la secciones anteriores se aplica normalmente a un único problema computacionalmente costoso siguiendo el paradigma *divide y vencerás*. Sin embargo, los problemas de VS tratados en esta tesis representan justamente lo contrario a un problema resuelto mediante paralelismo clásico. En LBVS, encontrar la mejor solución dado un compuesto con otro de referencia apenas necesita unas centésimas de segundo por lo que paralelizar estos problemas es contraproducente. Sin embargo, si bien no se puede paralelizar cada una de las comparaciones, si se puede dividir la base de datos en varias partes y procesar cada una por separado y de forma paralela. En consecuencia, en esta tesis se ha aplicado un paralelismo mediante balanceo de carga aprovechando la arquitectura MIMD de los computadores y su memoria distribuida.

A continuación se va a explicar la metodología que se ha realizado para paralelizar y distribuir

las ejecuciones a través de los centros de computación. La figura 1.16 ilustra un ejemplo de una ejecución paralela en dos centros de computación de una molécula de referencia contra dos bases de datos utilizando para cada una un algoritmo diferente. Las etapas representadas en la figura se corresponden con las descritas a continuación:

1. Se reúne todo lo necesario en el equipo personal. En este primer paso se deben de procesar la molécula query y las bases de datos para que los experimentos se realicen bajos las mismas condiciones. Para los ejecutables de los algoritmos hay que considerar el hardware y plataforma sobre la que se realizarán los experimentos.
2. Se divide el problema en dos, pues disponemos de dos centros de supercomputación en este caso en particular. En consecuencia, cada base de datos y algoritmo se reparte y la molécula query se duplica para los dos problemas.
3. Una vez se conocen las características de cada subproblema, se generan los scripts. El número y contenido de cada script dependerá del tamaño de la base de datos y de las particiones en las que se ejecutarán. En ocasiones, algunos compuestos suelen dar resultados erróneos u obligan a detener la ejecución del algoritmo ya sea por problemas del propio compuesto o, normalmente, problemas del sistema de archivos del centro de computación. Para reducir el número de experimentos a repetir, se crea un número de scripts de forma que cada script tenga el suficiente trabajo para mantener el nodo de la partición trabajando durante un tiempo correcto pero sin ser ni ejecuciones instantáneas ni que se prolonguen en el tiempo indefinidamente. Si los scripts se ejecutan demasiado rápido y otro usuario ha solicitado el mismo nodo, deberemos esperar a que terminen sus trabajos para seguir lanzando los nuestros. Pero si se crea un único script con toda la experimentación, además de arriesgar la parada de todos los experimentos por un error en un compuesto, normalmente existen restricciones de uso en las particiones en los centros para que un usuario no consuma únicamente todos los recursos de un nodo. En consecuencia, hay que encontrar un equilibrio.
4. Tras la generación de los scripts, se envían junto a las bases de datos, los ejecutables de los algoritmos y las queries a sus respectivos centros de cómputo.
5. Desde el nodo de control se lanzan los scripts, los cuales inicialmente se ponen en cola en el sistema Slurm. Cuando el hardware requerido esté disponible, se ejecutarán los experimentos almacenándose sus resultados en los directores que se hayan configurado para tal fin.
6. Una vez se ha ejecutado todo, o al menos alguno de los subproblemas de forma completa, se recuperan los resultados con el ordenador personal. Este es uno de los pasos más críticos pues es el momento en el que se obtienen los resultados y pueden verse los problemas que hayan podido suceder. Además, en este etapa, en las primeras ejecuciones de un problema, se suele obtener una muestra parcial de los resultados para comprobar la reproducibilidad de los mismos. Esto se realiza manualmente. Cabe mencionar que para este ejemplo se ha reducido considerablemente el número de scripts representados pero en los casos prácticos el número asciende a cientos de scripts que deben agrupar sus resultados en unos pocos archivos resultantes. Estos archivos serán los que lleguen finalmente al ordenador personal.
7. El último paso, tras obtener los resultados finales en el ordenador personal, consiste en generar el documento con únicamente la información necesaria que permita obtener las conclusiones de esos resultados. En esta etapa, al igual que en la anterior, las primeras ejecuciones de un nuevo problema se suele comprobar minuciosamente para asegurar que el proceso es correcto.

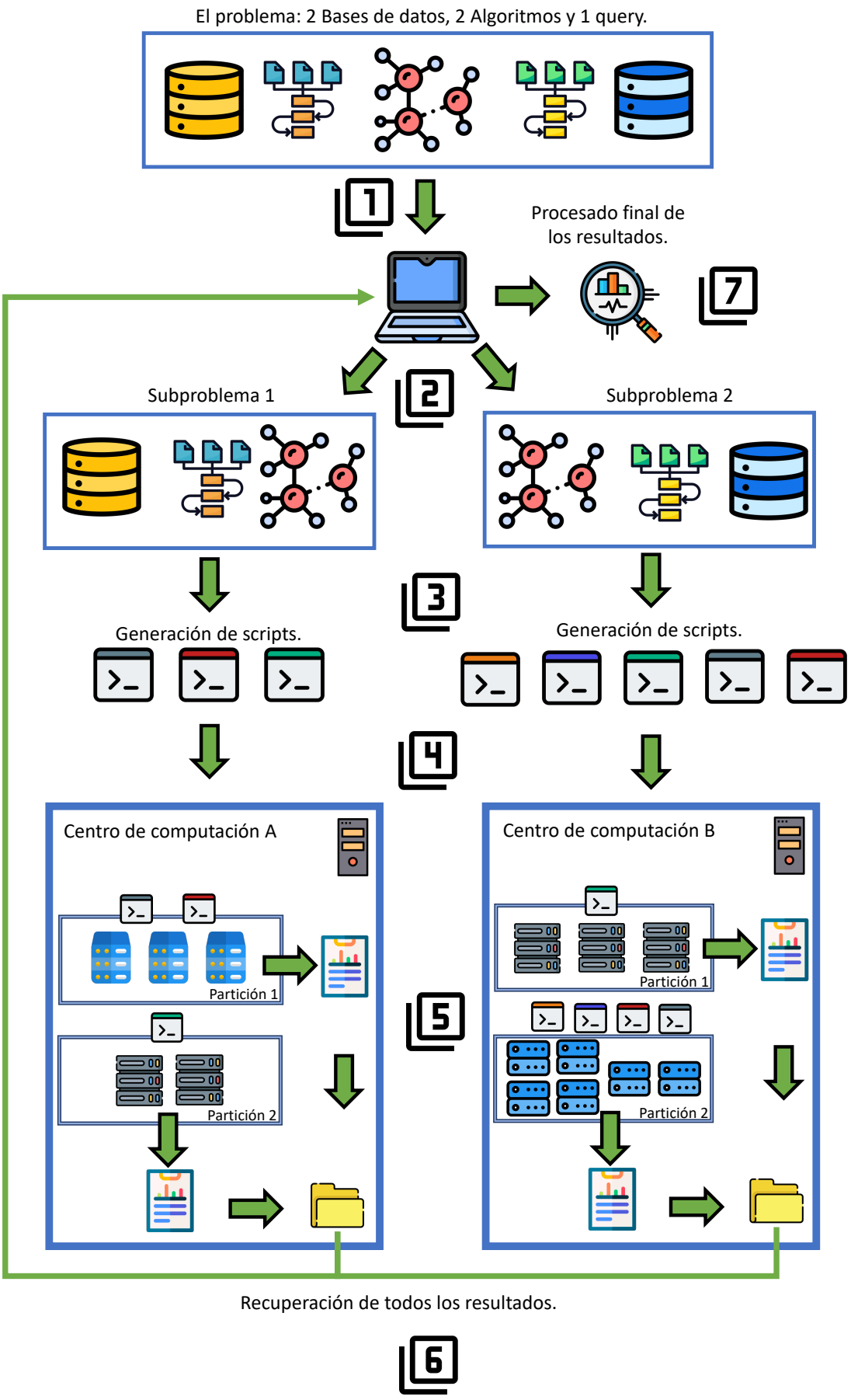


Figura 1.16: Ejemplo de balanceo de carga de un problema de LBVS mediante el reparto de trabajo según las especificaciones del problema y características de los centros de computación.

2. OptiPharm

En este capítulo se define y explica OptiPharm [107]. OptiPharm es un algoritmo o método diseñado para hacer LBVS. Su característica principal es que no está limitado a un único descriptor, como ocurre en la mayoría de los métodos propuestos en la literatura, sino que puede resolver cualquier problema de optimización que implique el cálculo de la similitud de dos compuestos dados como parámetros de entrada. En otras palabras, es independiente de la función objetivo utilizada para medir la similitud entre dos moléculas dadas. OptiPharm es un método de optimización global en el sentido de que analiza todo el espacio de búsqueda con el fin de encontrar áreas prometedoras donde puedan estar los óptimos locales y globales. No obstante, en cada una de estas áreas prometedoras, OptiPharm aplica mejoras u optimizaciones locales utilizando la información propia del problema. Todo ello con el objetivo de maximizar la similitud entre una molécula de referencia o molécula fija, denominada query, y una molécula variable de la base de datos denominada target. Las distintas mejoras del optimizador consisten en ajustes graduales de la molécula target con la query. En consecuencia, al utilizar optimizadores locales dentro del proceso de optimización, OptiPharm se clasifica dentro de los algoritmos meméticos. Además, OptiPharm puede considerarse un algoritmo de propósito general, en el sentido de que puede utilizarse para resolver cualquier problema de optimización que implique el cálculo de la similitud de dos compuestos dados como parámetros de entrada. En otras palabras, es independiente de la función objetivo utilizada para medir la similitud entre dos moléculas dadas.

El capítulo se estructura de la siguiente forma. En la sección 2.1 se describen los parámetros de optimización de OptiPharm y en la sección 2.2 se detallan los métodos de OptiPharm.

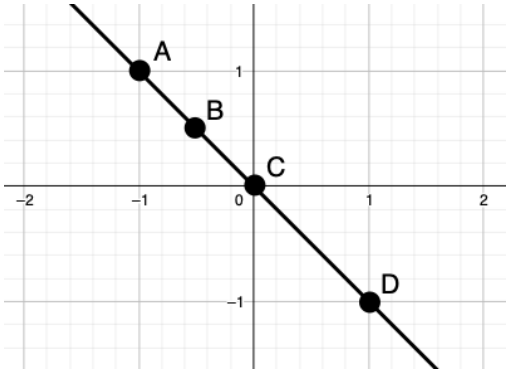


Figura 2.1: Ejemplo en el que se puede observar como dos pares de puntos distintos pueden formar un mismo eje.

2.1 Parámetros de optimización

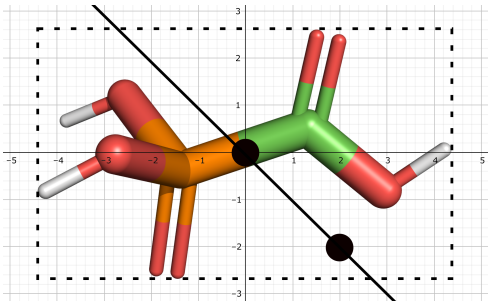
En OptiPharm, una solución $\boldsymbol{s}$ representa la rotación y la traslación que debe realizar un compuesto target para ser posteriormente evaluada con una función objetivo. En concreto, $\boldsymbol{s}$ es un cuaternión de la forma $\boldsymbol{s} = (\theta, \boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{\Delta})$, donde θ es el ángulo de rotación que se aplicará sobre un eje de rotación definido, en un sistema de coordenadas euclideo, por los puntos $\boldsymbol{c_1} = (x_1, y_1, z_1)$ y $\boldsymbol{c_2} = (x_2, y_2, z_2)$. $\boldsymbol{\Delta} = (\Delta x, \Delta y, \Delta z)$ representa un vector de desplazamiento.

Los parámetros asociados al cuaternión $\boldsymbol{s}$ se encuentran delimitados en unos rangos establecidos, y sus valores dependen de los compuestos de entrada. El intervalo en el que se encuentra el ángulo de rotación θ es independiente del tamaño de los compuestos y su valor siempre está comprendido en el intervalo $[0, 2\pi)$, por tanto, se puede realizar cualquier ángulo de rotación sobre un eje dado. Sin embargo, hay que tener en cuenta que las moléculas pueden tener diferentes tamaños. Por ese motivo, buscando reducir el espacio de búsqueda y, por tanto, el tiempo requerido para la optimización, OptiPharm calcula dinámicamente los límites correspondientes para los parámetros $\boldsymbol{c_1}$ y $\boldsymbol{c_2}$. Más concretamente, los límites se establecen en los puntos extremos de una de las diagonales del paralelepípedo que contiene a la molécula target. Es importante mencionar que este eje puede venir dado por un número infinito de dos coordenadas. En la figura 2.1 puede apreciarse que cualquier combinación de dos puntos, definen el mismo eje de rotación. Para evitar que el algoritmo explore soluciones equivalentes y aumentar así su eficiencia, se ha normalizado el eje. Una vez definidos los límites de los puntos que forman el eje de rotación, se pueden seguir dos técnicas para su generación:

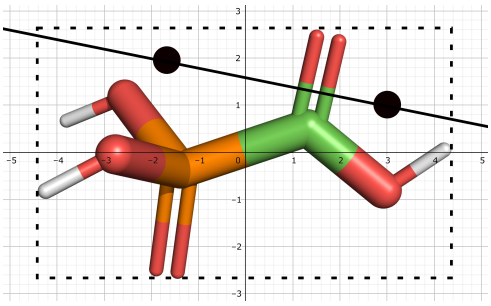
- Considerar un punto en el centro de masas de la molécula aprovechando que este punto se corresponde también con la coordenada $(0,0,0)$ y permitir que el otro se genere en cualquier lugar del espacio, siempre que esté dentro de los límites.
- Ambos puntos se generan de forma aleatoria dentro de los límites.

En la figura 2.2 se pueden visualizar las dos técnicas comentadas anteriormente para la creación de los ejes. Dependiendo del problema a resolver se puede utilizar una configuración u otra.

Finalmente, el intervalo de $\boldsymbol{\Delta}$ se establece en $[-\boldsymbol{maxD}, \boldsymbol{maxD}]$, siendo $\boldsymbol{maxD}$ la diferencia máxima entre las cajas que contienen a las moléculas query y target. A continuación

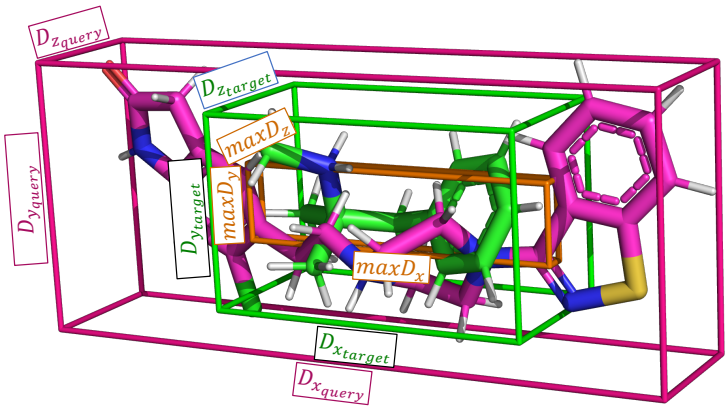


(a) Un punto es aleatorio y el otro se sitúa en el centro de masas de la molécula.



(b) Los dos puntos son generados aleatoriamente dentro de la caja.

Figura 2.2: Técnicas de generación de puntos para el eje de rotación.



$maxD_x$ = abs($D_{x_{target}}$ - $D_{x_{query}}$) $maxD_y$ = abs($D_{y_{target}}$ - $D_{y_{query}}$) $maxD_z$ = abs($D_{z_{target}}$ - $D_{z_{query}}$)

Figura 2.3: Cálculo de límites para la traslación de la molécula target.

se explica el proceso para su cálculo. En la figura 2.3 se pueden visualizar las tres cajas, las que engloban a las moléculas y la que delimita el desplazamiento. En primer lugar se centran ambos compuestos en el origen de coordenadas en base a su centro de masas. Posteriormente se calculan las cajas que contienen a cada molécula. Se ha representado en color verde la molécula query y de color rosa la molécula target así como sus respectivas cajas. Posteriormente, se restan las distancias entre los distintos ejes por separado ($\boldsymbol{maxD_x}, \boldsymbol{maxD_y}, \boldsymbol{maxD_z}$). Finalmente se selecciona para cada eje, el valor mayor que pasa a formar parte de $\boldsymbol{maxD}$. Esto da lugar a la caja naranja. Este es el desplazamiento desde el centro de la caja rosa que puede desplazarse el compuesto. Este procedimiento tiene como objetivo evitar que se generen soluciones $\boldsymbol{s}$ donde no exista superposición entre compuestos durante el proceso de optimización. En la figura 2.4 se muestra un ejemplo de una situación que se evita mediante este método.

2.1.1 Evaluación de una solución candidata

A continuación se explica como se realiza la rotación y traslación de una molécula, definida por una solución $\boldsymbol{s}$, para su posterior evaluación. Este procedimiento lo hemos ilustrado a partir de la figura 2.5, donde se consideran dos moléculas de entrada, una estructura cristalina deno-

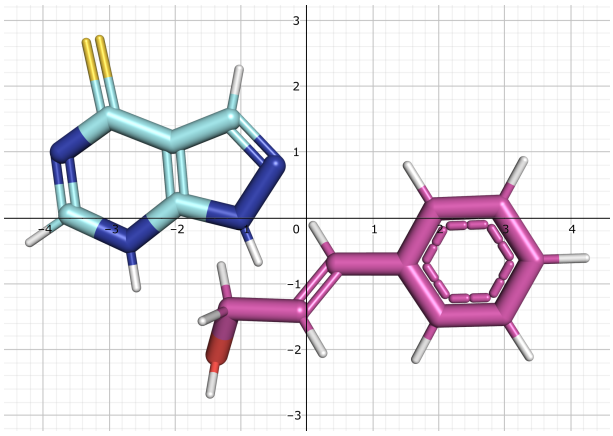


Figura 2.4: La correcta delimitación del parámetro Δ impide situaciones de baja superposición como la considerada en esta figura.

minada fxa (3KL6)[1], y dibujada en verde y un compuesto target denominado ZINC03819415[2] representado en fucsia. A modo de ejemplo hemos considerado como función de evaluación la similitud en forma Tc_S definida en la sección 1.1.2.1.

Los compuestos de entrada se encuentran inicialmente centrados en el centro de coordenadas (figura 2.5-1). La estructura cristalina fxa no se modificará durante todo el proceso. Por su parte, como el compuesto ZINC03819415 es el que se modificará, con el fin de poder apreciar cada cambio, en la imagen se representarán las resultantes orientaciones con colores diferentes. Así en las figuras 2.5-2 y 2.5-3 se representa la orientación de la molécula antes y después de la modificación en la orientación espacial. Los valores de los parámetros para esta solución son $s = [3, -3, 8.7, -1.5, 0, 0, 0, 1, 1.75, -1]$. El valor de Tc_S inicial es 0.18.

El primer paso consiste en la definición del eje y la rotación en torno al compuesto target (figura 2.5-2). Para ello se utilizan los 7 primeros parámetros de s, $[3, -3, 8.7, -1.5, 0, 0, 0]$ que definen un giro de 3 radianes sobre el eje formado por los puntos $c_1 = (-3, 8.7, -1.5)$ y $c_2 = (0, 0, 0)$. 3 radianes implicaría una rotación de 171º grados, de ahí que la posición resultante (compuesto amarillo) esté girado casi completamente respecto del original. Este es el proceso más costoso por lo que para reducir el tiempo necesario para rotar todos los átomos del compuesto target se han utilizado los cuaterniones, creados por W.R. Hamilton en 1843 [108].

El último paso es trasladar el compuesto según los últimos tres parámetros de s, $\Delta = (1, 1.75, -1)$, esto es desplazar 1 unidad en el eje X, 1.75 en el Y y -1 en el Z el compuesto target (figura 2.5-3). La nueva posición del compuesto target está representado en color salmón.

Finalmente, en la figura 2.5-4 se puede ver la posición final del compuesto target sobre la query con una similitud de forma de 0.53.

[1] `https://www.rcsb.org/structure/3kl6`
[2] `http://zinc.docking.org/substances/ZINC000003819415/`

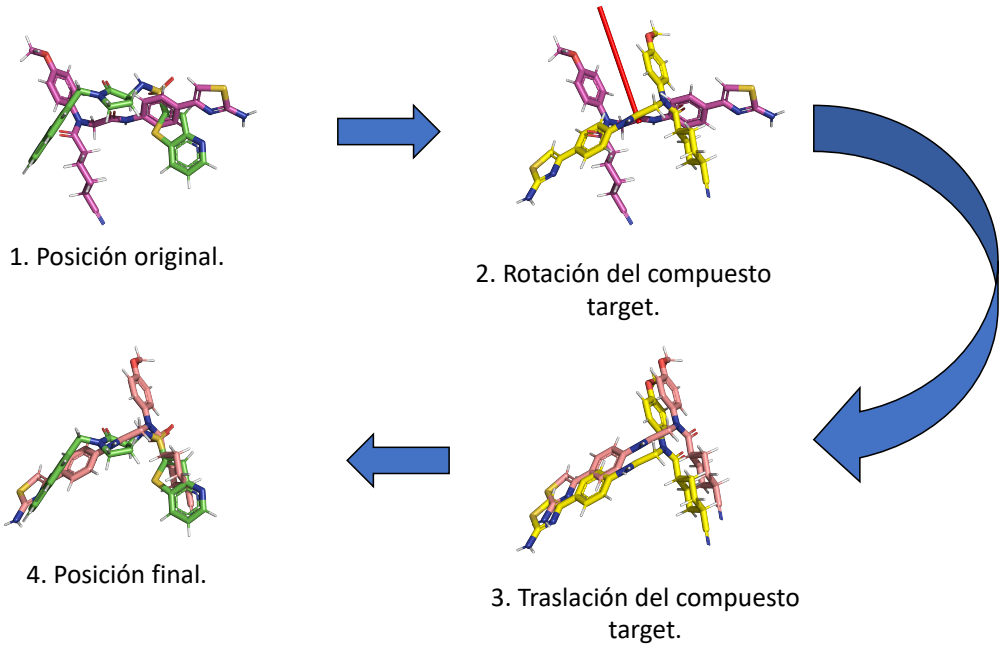


Figura 2.5: Proceso de evaluación de dos compuestos de entrada desde su posición inicial hasta la final a través de los parámetros s.

2.2 Algoritmo

OptiPharm es un nuevo algoritmo memético diseñado para resolver problemas de optimización global que incluye mecanismos para detectar subáreas prometedoras del espacio de búsqueda y descartar aquellas en las que no se espera encontrar óptimos globales. En otras palabras, OptiPharm intenta detectar otras nuevas que tengan el potencial de contener en óptimos locales o globales. Para hacerlo, OptiPharm inicialmente trabaja en un conjunto de M soluciones que se definen como población. Las soluciones pueden considerarse como puntos de partida independientes en los que OptiPharm aplica procedimientos de reproducción basados en la evolución natural, algo característico de los algoritmos evolutivos y meméticos. Que sean independientes significa que cada una de ellas puede encontrar nuevas soluciones descendientes en diferentes regiones prometedoras sin necesidad de relacionarse con el resto de la población. Luego, de entre todas las soluciones existentes, las M soluciones con mayor valor de similitud se mantendrán para la siguiente etapa, donde se mejoran mediante un optimizador local que es una herramienta que utilizan algunos algoritmos meméticos para permitir una mejor explotación del área de búsqueda. Esta secuencia de reproducción-reemplazo-mejora se repite tantas veces como se haya indicado en el parámetro t_{max} (ver figura 2.6).

Pero lo que realmente caracteriza a OptiPharm es el concepto de radio: todas las soluciones incluidas en la población tiene un radio asociado. Este delimita una subárea de n dimensiones dentro del espacio de búsqueda. A nivel conceptual se puede entender como una ventana en la que se aplican métodos de reproducción y optimización. El valor del radio asignado a una solución depende de la iteración i en la que la solución se haya generado. Más concretamente, el radio R_i de una nueva solución, encontrada durante el procedimiento de reproducción en la iteración i, viene definido por una función exponencial que disminuye a medida que aumenta el nivel o iteración. Esta función depende de dos valores: (i) del radio de las soluciones iniciales (el

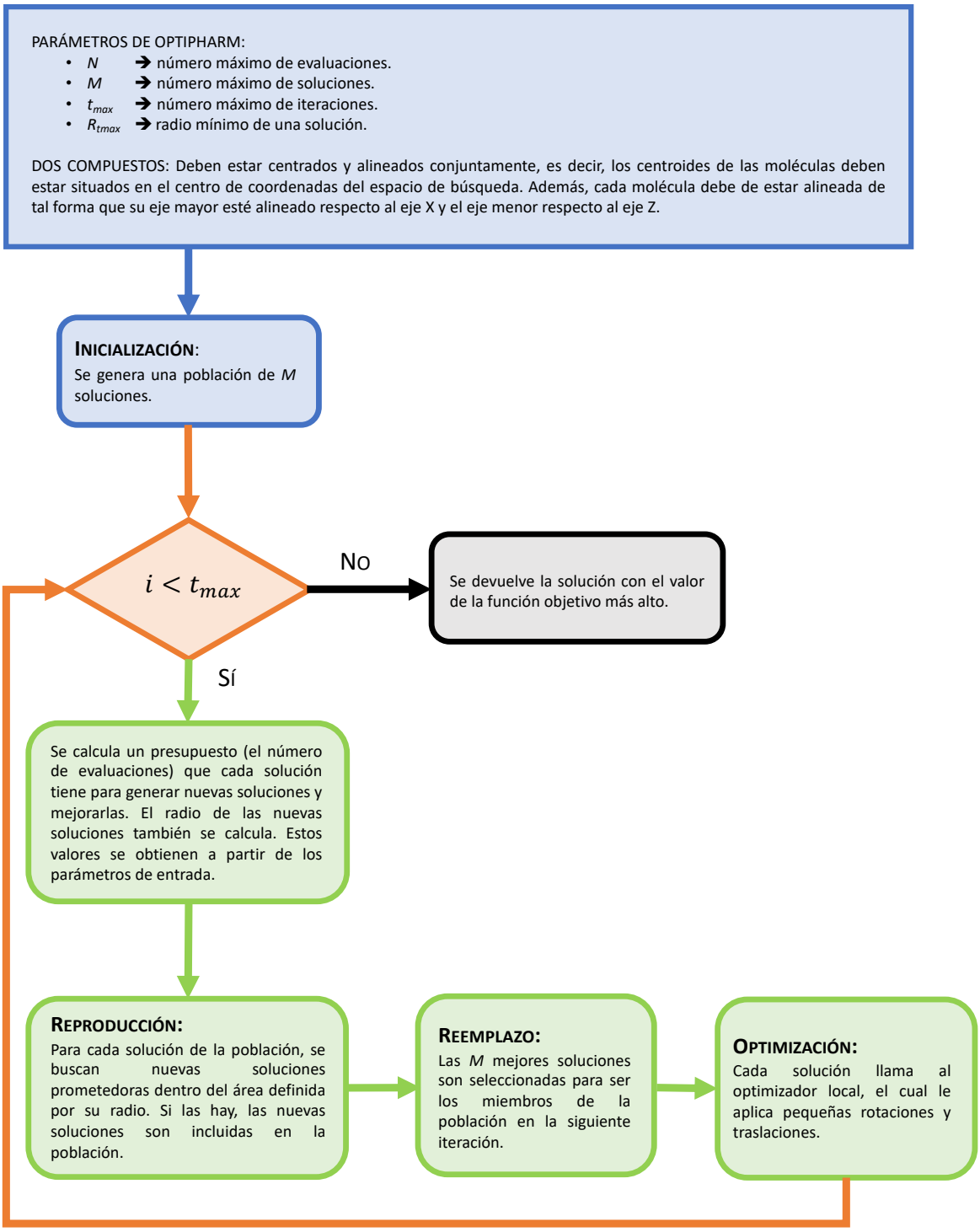


Figura 2.6: Estructura del algoritmo OptiPharm.

radio en el primer nivel, R_1) que tiene un valor que cubre todo el espacio de búsqueda y por tanto depende de este, y (ii) del radio de la soluciones en el último nivel definido por $R_{t_{max}}$, que se proporciona como parámetro de entrada. Este mecanismo de radio, el cual ha sido heredado de UEGO [109], está diseñado pensando en conseguir un equilibrio entre exploración y explotación.

Durante la ejecución de OptiPharm, pueden coexistir de forma simultanea varias soluciones candidatas con diferentes radios. Esto significa que el método puede analizar tanto subregiones

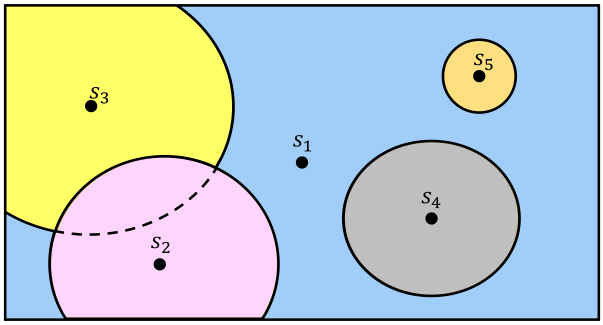


Figura 2.7: En OptiPharm pueden coexistir de forma simultánea varias soluciones con diferentes radios. Esta figura muestra un ejemplo para un caso de dos dimensiones.

grandes como pequeñas dentro de una misma etapa del procedimiento de optimización (ver figura 2.7).

Además de los parámetros ya mencionados como son el número máximo de soluciones iniciales M, el número de niveles o iteraciones t_{max} y el valor de radio en el último nivel $R_{t_{max}}$, OptiPharm requiere de otro parámetro de entrada adicional. Este es el número máximo de evaluaciones que realizará, N. Estas evaluaciones se distribuyen entre las soluciones candidatas en cada iteración, de tal forma que cada solución tenga un presupuesto para generar nuevas soluciones y mejorarlas. Estos presupuestos se calculan mediante ecuaciones que dependen de los parámetros de entrada mencionados anteriormente. Igual que el concepto de radio, esta idea también ha sido heredada de UEGO [109].

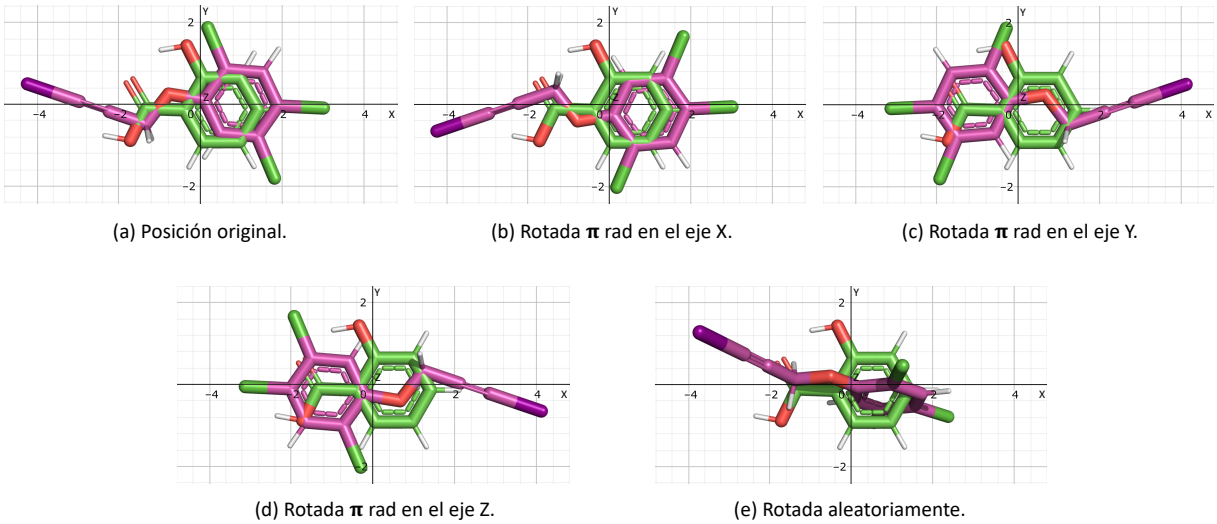
Los efectos de los posibles valores de los parámetros de UEGO y por tanto de OptiPharm fueron estudiados en un trabajo anterior [110]. Además, se dieron recomendaciones para ajustar los parámetros según el problema a resolver.

Finalmente, es necesario destacar que a diferencia de muchas de las heurísticas que se pueden encontrar en la literatura, los criterios de terminación de OptiPharm no se basan en el número de evaluaciones N, sino en el número de iteraciones t_{max}, de modo que el algoritmo puede terminar sin haber consumido el número máximo de evaluaciones permitido. El algoritmo se adapta a la complejidad del problema de modo que si la función objetivo tiene muchos óptimos y son de difícil acceso, realizará más evaluaciones que en el caso de funciones con pocos óptimos a los que se puede converger de forma no muy abrupta.

Una vez explicado de manera general OptiPharm, a continuación se detallarán cada una de sus etapas claves.

2.2.1 Método de inicialización

En la fase de inicialización, las dos moléculas de entrada están alineadas y centradas en el origen de coordenadas. Para ello se ha utilizado el Análisis Principal de Componentes (PCA, *Principal Component Analysis*), un método matemático que se utiliza para reducir el número de dimensiones o detectar aquellas que son principales de un conjunto de datos [111]. Aplicado a



(a) Posición original. (b) Rotada π rad en el eje X. (c) Rotada π rad en el eje Y.

(d) Rotada π rad en el eje Z. (e) Rotada aleatoriamente.

Figura 2.8: Soluciones iniciales para un caso de $M = 5$: (a) s_1, solución inicial; (b) s_2, obtenido con la rotación de s_1 π rad sobre el eje X; (c) s_3, obtenido con la rotación de s_1 π rad sobre el eje Y; (d) s_4, obtenido con la rotación de s_1 π rad sobre el eje Z; (e) s_5, todos los parámetros $(\theta, c_1, c_2, \Delta)$ son generados aleatoriamente dentro de los límites calculados por OptiPharm, para esta instancia en particular.

nuestro problema, los puntos se corresponden con la posición de los átomos en el espacio y el PCA ayuda a identificar los ejes principales para que estos sean alineados con los ejes de coordenadas X, Y y Z. Luego, a partir de esta situación inicial, se compone una población de M soluciones con $M >= 4$. La primera solución representa la posición inicial de los compuestos, es decir, la posición alineada de ambas moléculas. La solución con la que se representa dicho estado es igual a $s_1 = (\theta, c_1, c_2, \Delta) = (0, (0,0,0), (0,0,0), (0,0,0))$. Las tres soluciones siguientes representan la rotación de la molécula target π radianes en cada eje (siempre desde el estado inicial), dando como resultado las siguientes soluciones candidatas $s_2 = (\pi, (1,0,0), (0,0,0), (0,0,0))$, $s_3 = (\pi, (0,1,0), (0,0,0), (0,0,0))$ y $s_4 = (\pi, (0,0,1), (0,0,0), (0,0,0))$. Finalmente, para evitar una deriva hacia los óptimos locales e introducir algo de aleatoriedad, se incluyen $M-4$ soluciones con todos sus parámetros generados aleatoriamente. La figura 2.8 muestra un ejemplo de las soluciones iniciales para un caso en particular con $M = 5$. Como se puede observar, siempre existe superposición entre ambos compuestos.

2.2.2 Método de reproducción

El método de reproducción es el encargado de explorar el espacio definido por el radio de cada solución s de la población (ver figura 2.7). El objetivo que persigue este método es encontrar nuevas soluciones prometedoras que evolucionen hacia óptimos locales o globales en las fases posteriores del algoritmo. Cada solución es independiente del resto por lo que su espacio se analiza también de manera independiente. El proceso es el siguiente:

De cada solución s_i en la población, las nuevas soluciones candidatas s_{ij} se generan aleatoriamente en el área definida por su radio (ver figura 2.9a). Además, para cada par de soluciones generadas (s_{ij} y s_{ik}), se obtiene el punto medio del segmento ($Mid(s_{ij}, s_{ik})$) que conecta al par de puntos generados (ver figura 2.9b). A continuación, se calcula el valor de la función objetivo de los puntos extremos ($f(s_{ij})$ y $f(s_{ik})$), así como también el del punto medio

($f(\boldsymbol{Mid}(\boldsymbol{s_{ij}}, \boldsymbol{s_{ik}}))$). Si alguno de estos nuevos puntos tiene mejor valor de la función objetivo que la solución original $\boldsymbol{s_i}$, esta se reemplazará por dicho punto, es decir, el centro de esa subárea $\boldsymbol{s_i}$ pasará a ser el que mejor valor de la función objetivo tenga. Además, si el punto intermedio tiene un valor de la función objetivo peor que el de los puntos extremos, esto puede significar que el punto intermedio se encuentra en un valle (ver figura 2.9b) y por consiguiente los puntos extremos pueden estar en diferentes colinas y se insertarán en la lista de la población como nuevas soluciones candidatas. El radio de estos puntos generados será el asociado con la iteración actual. La figura 2.9c muestra un resumen de todo el proceso manteniendo las referencias a los nombres en las figuras 2.9a y 2.9b.

2.2.3 Método de reemplazo

Tras aplicar el método de reproducción, posiblemente el tamaño de la población sea mayor que el valor del parámetro de entrada *M* que define el tamaño máximo permitido de la población. En consecuencia, se debe aplicar un procedimiento para seleccionar solo las soluciones supervivientes. Existes diferentes tipos de sustituciones. En este trabajo se ha implementado uno altamente elitista y determinista que consiste en agrupar a la población original y los descendientes en una población intermedia para luego seleccionar las *M* mejores soluciones. Estos serán los que pasen a la siguiente etapa. El resto se eliminan.

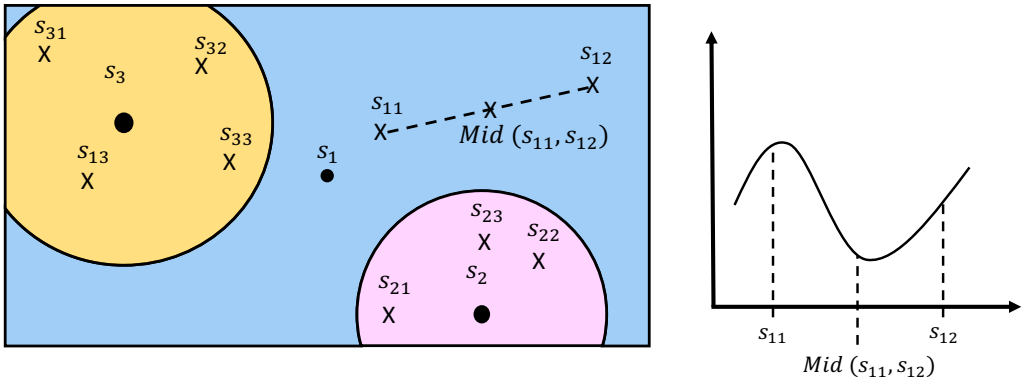
Implícito en este método de sustitución se encuentra un procedimiento de clasificación que ordena las soluciones en base a sus valores de la función objetivo.

2.2.4 Método de mejora

Para evitar la convergencia hacia los óptimos locales e introducir algo de ruido en el proceso de búsqueda, en los algoritmos evolutivos y genéticos se aplica un operador de mutación a la población actual que se suele incluir en el procedimiento de optimización realizando pequeños cambios aleatorios en los individuos. Por su parte, en los algoritmos meméticos se aplica un algoritmo de búsqueda o algoritmo de mejora que puede consistir en un procedimiento de búsqueda local. Para los problemas abordados en esta tesis, el uso de estos últimos ha demostrado una mejor aproximación de las soluciones hacia los óptimos.

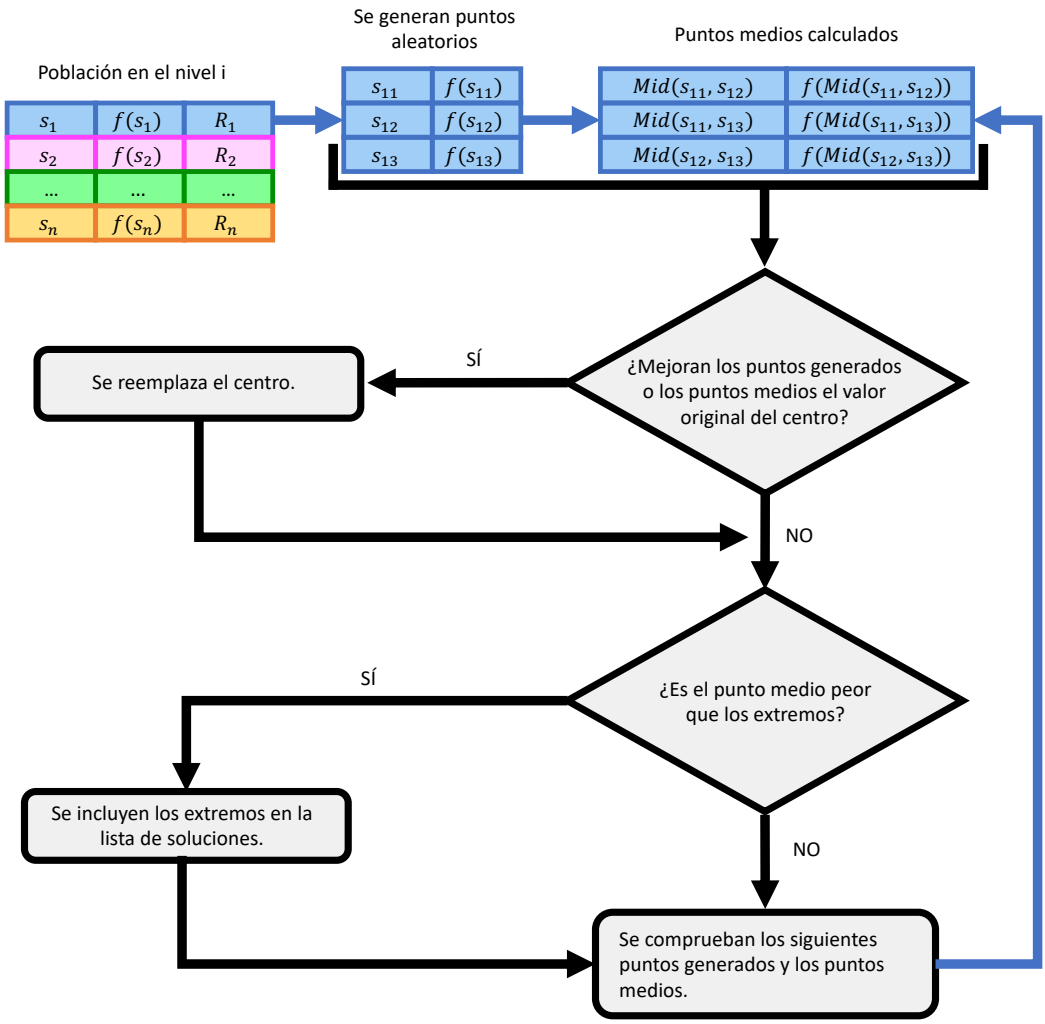
El método de mejora implementado en OptiPharm es el método de búsqueda local SASS, inicialmente propuesto por Solis y Wets [58]. Se ha elegido principalmente porque es un algoritmo de optimización heurístico que se puede aplicar para maximizar cualquier función arbitraria sobre un subconjunto acotado de $\mathbb{R}^N$. Este no se ha aplicado directamente sino que se ha adaptado al problema en cuestión. A continuación se detallan brevemente estos cambios.

El algoritmo SASS trabaja internamente que el rango $[0,1]$ para cada una de las variables. Nuestros problemas no están delimitados en este rango por lo que ha sido necesario usar una función para reescalar los valores de las variables al intervalo $[0,1]$, y una función de desnormalización para invertir dicho proceso. En SASS, los nuevos puntos se generan utilizando una perturbación gaussiana $\xi \in \mathbb{R}^3$ sobre el punto de búsqueda (x, α) y un término de sesgo normalizado $b \in \mathbb{R}^3$ para dirigir la búsqueda. La desviación estándar σ especifica el tamaño de la esfera que probablemente contiene el vector de perturbación. En esta tesis, su límite superior σ_{ub}



(a) Espacio de búsqueda en 2D

(b) Espacio imagen en 2D



(c) Procedimiento para generar nuevas soluciones.

Figura 2.9: Método de reproducción.

debe tener el mismo valor que el radio normalizado de la solución que ejecuta el método. Por otra parte, el parámetro σ_{ub} se considera un argumento de SASS. Lo que permite esa acotación es que los pasos que da el optimizador nunca son más largos que el valor del radio de la solución candidata. Finalmente, la optimización finaliza cuando el número máximo de evaluaciones($f_{e_{\text{máx}}}$) y/o el número máximo de fallos consecutivas (Max_{fcnt}) se alcanzan.

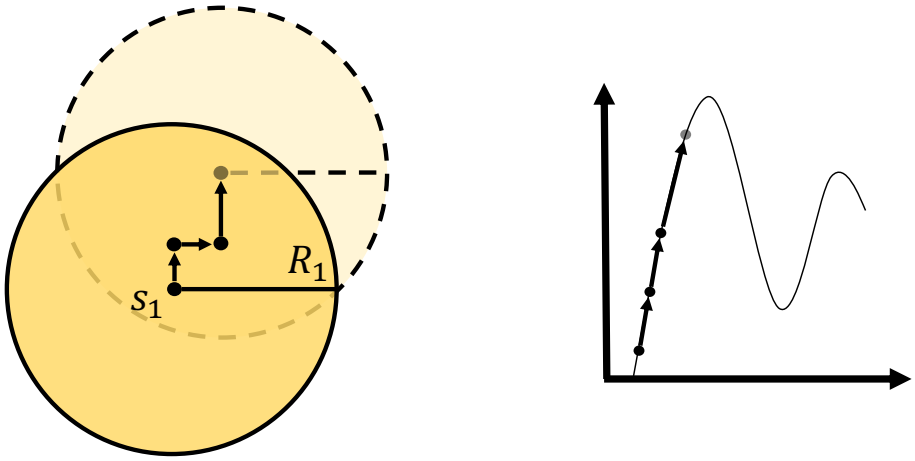


Figura 2.10: Ejemplo en 2D del optimizador local SASS. SASS busca una dirección de mejora y mueve el centro de la solución a través de esa dirección dando saltos de diferentes tamaños. Si se van consiguiendo de forma consecutiva mejores valores de la solución, los saltos son cada vez más largos, teniendo como límite el radio de esa solución. Si por el contrario, no se produce mejora con los saltos, estos se irán reduciendo. Finalizará el proceso de optimización cuando se alcance el número máximo de evaluaciones y/o el número máximo de fallos consecutivos.

OptiPharm aplica el SASS a cada solución de la población. En la figura 2.10 se puede ver un ejemplo de su funcionamiento.

II Aplicaciones

3. LBVS basado en la similitud de forma

En este capítulo se analiza la eficiencia y eficacia de OptiPharm para abordar el problema de LBVS basado en la forma de los compuestos. En la sección 3.1 se define el problema de optimización a resolver y se resumen los enfoques previos de la literatura. En la sección 3.2 se definen las condiciones bajo las que se han realizado los experimentos indicando la configuración de los algoritmos y las bases de datos utilizadas. En la sección 3.3 se muestra la capacidad de OptiPharm para encontrar soluciones buenas utilizando la base de datos Maybridge. Posteriormente en las secciones 3.4 y 3.5 se realiza una comparativa entre OptiPharm y WEGA (el actual estado del arte) sin considerar los hidrógenos usando las bases de datos FDA, DUD y DUD-E. Los últimos experimentos se muestran en la sección 3.6 donde se ha realizado un estudio sobre la importancia de considerar los átomos de hidrógeno en los compuestos durante las comparaciones. Finalmente la sección 3.7 recoge las conclusiones de este capítulo.

3.1 Problema de optimización

En el LBVS basado en la forma molecular, los compuestos de una gran base de datos se comparan con uno de referencia para proporcionar información sobre cuáles de las moléculas de la base de datos tienen una alta similitud en forma con respecto al compuesto de referencia. La función Tc_S correspondiente al modelo gaussiano definido en la sección 1.1.2.1 será la utilizada para este problema de optimización denominado como P_{Sqt} y definido matemáticamente en el problema 3.1.

Problema 3.1

$$P_{Sqt} = \begin{cases} \text{máx} & Tc_{Sqt}(\Theta_t, c1_t, c2_t, \Delta_t), \\ \text{s.t.} & \Theta_t \in [0, 2\pi](\text{rad}), \\ & c1_t, c2_t \in [(-box_{t_x}, -box_{t_y}, -box_{t_z}), (\overline{box_{t_x}, box_{t_y}, box_{t_z}})](\text{Å}), \\ & \Delta_t \in [(-\Delta_{qt_x}, -\Delta_{qt_y}, -\Delta_{qt_z}), (\overline{\Delta_{qt_x}, \Delta_{qt_y}, \Delta_{qt_z}})](\text{Å}) \end{cases} \tag{3.1}$$

P_{Sqt} se puede entender como un problema de maximización en el que dado un compuesto query q y una base de datos de compuestos DB, hay que encontrar para cada compuesto $t \in DB$ el máximo valor de similitud para cada par $q,t(Tc_{Sqt})$ optimizando las variables de decisión dentro de sus límites (Θ, c_1, c_2 y Δ) definidas en el capítulo 2. Estos límites se calculan dinámicamente para cada pareja de compuestos y dependen de los tamaños de estos últimos.

Una vez se conocen los valores de similitud para cada t respecto de q, se selecciona aquel que tenga el valor más alto obteniendo así el compuesto más similar:

$$BestComp = \text{máx}(P_{Sqt}) \; \forall t \in DB$$

Como solución al Problema 3.1, existe un algoritmo en la literatura llamado WEGA [29] que será el objeto de comparación de OptiPharm. WEGA fue el primer método que resolvió parcialmente algunos de los problemas de ROCS (considerado previamente el estado del arte) [112]. WEGA es un optimizador local concebido para maximizar la superposición entre dos moléculas dadas como parámetros de entrada. Comienza la búsqueda con una solución inicial y la desplaza de vecino en vecino mientras aumente el valor de la función objetivo Tc_S, definida por la ecuación 1.4. La principal ventaja de WEGA es su capacidad para encontrar una solución en un período de tiempo corto. Por el contrario, su principal inconveniente es su dificultad para escapar de los óptimos locales donde la búsqueda no puede encontrar ninguna solución vecina adicional que mejore el valor de la función objetivo, es decir, la calidad de la solución final depende estrechamente de la solución inicial considerada. Para hacer frente a este inconveniente y aumentar su probabilidad de éxito, WEGA considera más de un punto de partida. Más concretamente, aplica el optimizador local desde cuatro soluciones iniciales diferentes. La primera solución de partida consiste en las dos moléculas de entrada centradas y alineadas en el origen de coordenadas. Las restantes se obtienen girando el compuesto target π radianes (180 grados) en cada eje [29].

3.2 Configuración de los estudios computacionales

OptiPharm es un algoritmo memético evolutivo, por lo que para analizar su rendimiento, ejecutamos cada instancia varias veces y proporcionamos algunas métricas estadísticas, como se requiere cuando se evalúa cualquier algoritmo heurístico en la literatura [113-117]. Desde el punto de vista estadístico, se debe considerar un número mínimo de 30 repeticiones [118]. Sin embargo, los experimentos realizados se han ejecutado 100 veces para aumentar la confianza en los resultados. Con ellos, se calcula el valor medio y la desviación estándar para analizar su eficacia y eficiencia. Es importante destacar que ejecutar varias veces una instancia en particular es solo una metodología para analizar la robustez del algoritmo, pero en el escenario del mundo real, OptiPharm solo necesita una sola ejecución para proporcionar resultados confiables. En cuanto a WEGA, este es determinista, por lo que cada instancia se ejecutará una única vez.

El algoritmo subyacente de OptiPharm es parametrizable, lo que significa que puede ajustarse según las preferencias del usuario. Por lo tanto, en base a las necesidades del problema, se puede configurar para obtener soluciones de alta calidad a expensas de aumentar ligeramente el esfuerzo computacional o por el contrario, obtener una solución aceptable con un tiempo de computación

razonable. En este capítulo, los parámetros de OptiPharm (sección 2.2) se ajustaron probando varias combinaciones de valores de parámetros con un conjunto reducido de problemas y siguiendo las pautas descritas en un trabajo anterior [110]. Como consecuencia, se proponen dos conjuntos diferentes de parámetros de entrada, lo que da lugar a dos versiones de OptiPharm con diferentes objetivos:

(i) *OptiPharm Robust* (OpR). En este caso, el conjunto de parámetros de entrada se elige para hacer que OptiPharm sea confiable y robusto; en otras palabras, para permitir que OptiPharm explore y explote profundamente el espacio de soluciones en la búsqueda de la mejor solución posible. En particular, se consideraron los siguientes valores: $N = 200000$ evaluaciones, $M = 5$ soluciones iniciales, $t_{max} = 5$ iteraciones y $R_{t_{max}} = 1$ como el radio de las soluciones del último nivel.

(ii) *OptiPharm Fast* (OpF). En esta ocasión, los parámetros se ajustan para que los tiempos de ejecución sean más bajos o similares a los de WEGA, lo que permite una comparación equitativa entre ambos algoritmos. Se consideraron los siguientes valores: $N = 1000$ evaluaciones, $M = 5$ soluciones iniciales, $t_{max} = 5$ iteraciones y un radio mínimo de $R_{t_{max}} = 5$ para las soluciones del último nivel.

De los párrafos anteriores, se podría inferir que el número de soluciones iniciales, $M = 5$, y el número de iteraciones, $t_{max} = 5$, se pueden fijar independientemente del objetivo perseguido, mientras que el radio más pequeño $R_{t_{max}}$, y lo más importante, el número máximo de evaluaciones, N, tiene una mayor influencia tanto en la eficacia como en la eficiencia del algoritmo.

Las bases de datos que se han utilizado para comprobar la calidad de OptiPharm frente a WEGA son la Maybridge, FDA, DUD y DUD-E que se introdujeron en la sección 1.1.3.2. Hay que tener en cuenta que WEGA, a diferencia de OptiPharm, no considera los átomos de hidrógeno en los cálculos de similitud de forma. En consecuencia, OptiPharm, que se ha diseñado para que pueda considerar o no dichos átomos, para la comparativa con WEGA se han omitido. En relación al tamaño de los átomos, OptiPharm puede utilizar diferentes configuraciones. En este caso en el que se compara con WEGA se ha configurado de modo que todos los radios de todos los átomos tiene un valor de 1.7 Å [29]. Por último, los centroides de los compuestos de las bases de datos se han centrado en el eje de coordenadas y se han alineado de tal forma que el lado más largo está alineado con el eje X y el más corto con el eje Z.

En las siguientes secciones se presentan los distintos experimentos realizados. La sección 3.3 muestra un estudio para validar la capacidad de OptiPharm para encontrar soluciones de buena calidad cuando estas existen. En la sección 3.4 se compara WEGA con OptiPharm utilizando las bases de datos FDA sin hidrógenos. En la sección 3.5 se utilizarán las bases de datos DUD y DUD-E sin hidrógenos para comparar la capacidad de clasificación entre ambos algoritmos. Finalmente, en la sección 3.6 se analiza el efecto de incluir los hidrógenos de las moléculas aprovechando que OptiPharm permite considerarlos en sus evaluaciones. Además, es importante mencionar que en todos los experimentos se obtiene una puntuación igual a 1 cuando se compara una molécula consigo misma. Por lo tanto, cuando se mencione "la molécula con la mayor similitud de forma respecto de la query", denominada como *BestComp*, realmente se está excluyendo el caso en el que las moléculas query y target son iguales.

Tabla 3.1: Base de datos Maybridge. Se muestra el número *nC* de queries de la base de datos con un número de átomos $nA \in [i, j)$. De cada intervalo, se seleccionó un compuesto (query) al azar, y la molécula de la base de datos (*BestComp*) con el Tc_S más alto se calculó utilizando OpR. Tenga en cuenta que la puntuación Tc_S es igual a 1 cuando el compuesto de consulta se compara consigo mismo para todas las instancias, de modo que *BestComp* realmente representa la segunda molécula más similar a la consulta.

		Compuestos con $nA < 95$					Compuestos con $nA \geq 95$		
[i, j)	*nC*	query	*BestComp*	Tc_S	[i, j)	*nC*	query	*BestComp*	Tc_S
[0, 5)	0	-	-	-	[95, 100)	6	JFD01206	JFD01203	0.930
[5, 10)	2	CD08226	RF01682	0.940	[100, 105)	3	JFD00633	JFD01915	0.875
[10, 15)	93	AC10702	KM03331	0.982	[105, 110)	3	JFD02451	JFD02452	0.762
[15, 20)	968	AC10402	RF03315	0.939	[110, 115)	3	JFD01915	JFD00633	0.877
[20, 25)	3469	AC11546	NRB00891	0.940	[115, 120)	1	JFD02945	RH00477	0.512
[25, 30)	7050	AC10751	AC11968	0.991	[120, 125)	2	BTB14731	JFD01602	0.508
[30, 35)	10414	AC12586	RH01548	0.895	[125, 130)	1	JFD01714	JFD01716	0.676
[35, 40)	10623	AC10018	JFD00624	0.939	[130, 135)	0	-	-	-
[40, 45)	9015	AC10608	HTS01369	0.867	[135, 140)	1	JFD02946	RJC01701	0.474
[45, 50)	6085	AW00180	AW00174	0.873	[140, 145)	0	-	-	-
[50, 55)	3008	AW00136	HTS03294	0.849	[145, 150)	1	JFD02949	JFD00655	0.552
[55, 60)	1479	JFD00968	RJC02093	0.993	[150, 155)	2	BTB12204	BTB12205	0.600
[60, 65)	648	JFD03035	NRB03291	0.972	[155, 160)	2	BTB12205	BTB12204	0.600
[65, 70)	247	HTS13346	HTS13343	0.982	[160, 165)	1	RJC01719	BTB12214	0.487
[70, 75)	108	JFD01818	RJC03231	0.976	[165, 170)	2	RJC01701	JFD02451	0.645
[75, 80)	57	JFD01718	JFD01716	0.957	[170, 175)	0	-	-	-
[80, 85)	50	NRB03718	NRB03775	0.991	[175, 180)	0	-	-	-
[85, 90)	40	JFD00292	JFD00294	0.877	[180, 185)	1	JFD02950	JFD00655	0.417
[90, 95)	14	JFD01716	JFD01718	0.959	[185, 190)	0	-	-	-
media	53370			0.940		29			0.637

3.3 Calidad de las soluciones obtenidas por OptiPharm: Estudio computacional con la base de datos Maybridge.

Para comprobar que OptiPharm puede encontrar soluciones buenas cuando estas existen, se utilizó la base de datos Maybridge seleccionando un conjunto de compuestos queries que serían evaluados frente al resto de compuestos. El procedimiento de selección se llevó a cabo de la siguiente manera: los compuestos de la base de datos Maybridge se ordenaron inicialmente según su número de átomos y se dividió en 38 intervalos donde cada intervalo tenía un rango de 5 átomos más que el intervalo predecesor. Luego, se eligió al azar un único compuesto de cada intervalo. La tabla 3.1 resume los resultados obtenidos. En concreto se muestra: (i) el número de compuestos, *nC*, con una cantidad de átomos incluidos en el intervalo $nA \in$ [i, j); (ii) la query seleccionada al azar de dicho intervalo, y (iii) la molécula de Maybridge (*BestComp*) con el mayor valor de similitud de forma (Tc_S) según OpR. La última fila de la tabla muestra el número total de compuestos con $nA < 95$ (resp. $nA \geq 95$) y el valor medio de Tc_S. Se puede observar que existen intervalos con 0 compuestos, los cuales se han anotado con '-' en las filas correspondientes.

Como se puede ver en la tabla 3.1, OpR obtiene un valor de Tc_S medio de 0.940 para queries con nA < 95. Esto no es raro ya que la cantidad de compuestos con menos de 95 átomos es igual a 53370, por lo que la probabilidad de encontrar moléculas similares es relativamente alta. Por

el contrario, el valor medio de Tc_S obtenido por OpR para moléculas con más de 95 átomos es igual a 0.637, lo cual no es una mala cifra si consideramos que solo 29 de 53399 moléculas tienen más de 95 átomos. Aun así, OpR obtiene soluciones de buena calidad para queries con más de 95 átomos. Dos ejemplos de ello son las queries JFD0120 y JFD0063, con 96 y 104 átomos, respectivamente. Para estos dos casos, OpR ha encontrado compuestos con valores de Tc_S de 0.930 y 0.875, incluso cuando el número de moléculas con tamaños similares no es alto. Centrándonos ahora en los peores casos, es decir, aquellos en los que OpR obtiene los valores de Tc_S más bajos, como son los compuestos JFD02950 y JFD02946 con 180 y 135 átomos, respectivamente. Se puede observar que no hay moléculas en la base de datos con tamaños similares. En concreto, solo hay 10 moléculas, incluidas ellas mismas, con $nA \in [135, 190)$. Por lo tanto, la probabilidad de descubrir moléculas similares en términos de forma es muy baja, ya que lo más probable es que no existan. En consecuencia, a partir de los resultados, es posible inferir que OpR encuentra una solución de alta calidad para una query dada cuando existe en la base de datos moléculas de tamaño similar.

3.4 Comparativa de WEGA y OptiPharm con la base de datos FDA sin hidrógenos

En esta sección se compararán las soluciones obtenidas por el algoritmo de referencia WEGA con las obtenidas por OptiPharm, en sus dos versiones. Dado que WEGA no permite considerar los átomos de hidrógeno, OptiPharm se ha configurado de forma similar. De la misma forma, el radio de Van der Waals para todos los átomos es fijo para WEGA y su valor es 1.7 Å. OptiPharm ha sido configurado para tener los mismos valores y poder comparar los resultados en igualdad de condiciones.

La tabla 3.2 muestra, para cada query, su número de átomos (nA), el compuesto de la base de datos de la FDA con la mayor similitud de forma (*BestComp*) y la puntuación de función asociada (Tc_S), de acuerdo con OpR , OpF y WEGA. Como se puede ver, el algoritmo OpR proporciona los valores Tc_S más altos, aunque también es el método que consume más tiempo de acuerdo con la tabla 3.3. Esto significa que se pueden lograr mejores predicciones usando OpR cuando no hay limitaciones de tiempo. Sin embargo, si se requieren tiempos de ejecución más bajos, se deben considerar algoritmos como OpF o WEGA.

Hasta donde sabemos, no existe ningún algoritmo, método o programa que pueda proporcionar con certeza la molécula más similar a un compuesto query. Hasta esta tesis, WEGA era el algoritmo que proporcionaba los valores de similitud más altos [29, 119]. Ahora, como se puede ver en la tabla 3.2, OpR mejora a WEGA en términos de calidad de la solución, encontrando valores más altos de Tc_S al procesar un compuesto query contra la base de datos. Por lo tanto, para analizar la efectividad de OpF y WEGA en términos de sus predicciones, las soluciones proporcionadas por OpR se considerarán las óptimas.

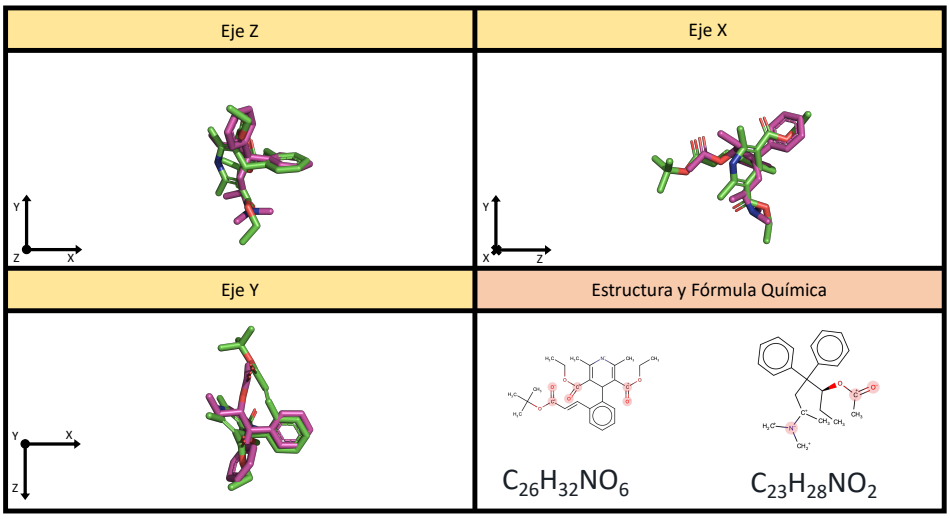
Como se puede ver en la tabla 3.2, las predicciones de WEGA coinciden con las de OpR en 22 de los 40 casos, mientras que OpF lo hace en 30 de 40. Esto representa una pequeña ventaja para OpF frente a WEGA en términos de éxito en las predicciones. Además, de la tabla 3.3, que muestra los tiempos de ejecución, se puede apreciar que OpF es más rápido que WEGA.

Asimismo, es importante estudiar los casos en que las predicciones de OpF y WEGA no coinciden con las logradas por OpR. Esto ocurre en 18 de 40 casos para WEGA y 10 veces para OpF. Para cada query, los 1751 compuestos se ordenan en orden descendente de acuerdo con el valor Tc_S obtenido por OpR. A continuación, se calcula la posición i en la lista donde está el *BestComp* alcanzado por OpF (resp. WEGA), y cuál es el valor de Tc_S de OpR. Esta información se muestra en la tabla 3.2 en las columnas 6 y 9 para OpF y WEGA, respectivamente. En términos generales, en la mayoría de los casos, las predicciones realizadas por OpF se encuentran en una mejor posición en la lista de OpR que las predicciones propuestas por WEGA.

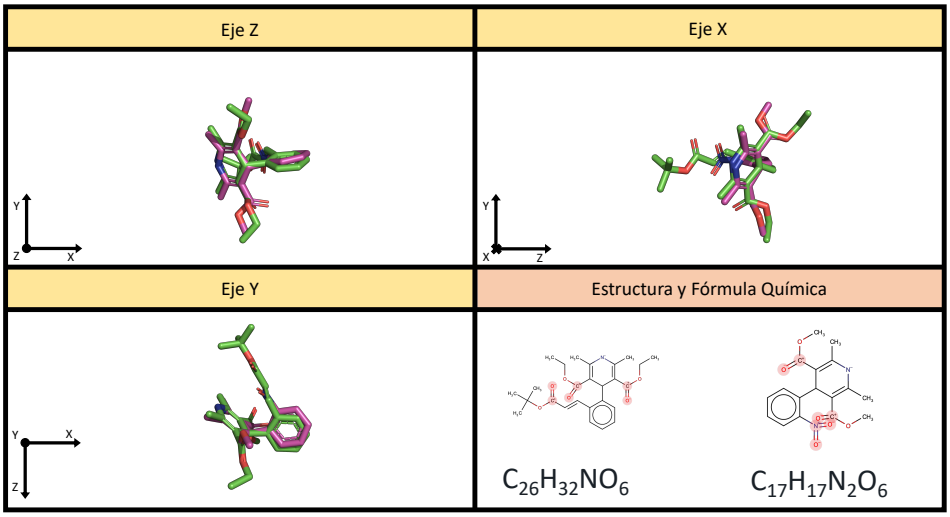
Es importante mencionar que, en general, OptiPharm está diseñado para mantener la diversidad de la población e investigar muchas soluciones candidatas en paralelo, evitando la deriva genética hacia una solución óptima única (local o global). Sin embargo, dependiendo del conjunto de parámetros seleccionado, la precisión al aproximarse a los óptimos puede ser mayor o menor. Por esta razón, OpF se ha ajustado para explorar el espacio de búsqueda en busca de las soluciones más prometedoras, pero sin perder tiempo "mejorándolas". En términos de optimización, los parámetros de entrada se seleccionan para determinar los picos más altos en el espacio de búsqueda, pero no para llegar realmente a la cima del pico más alto. Incluso cuando OpF propone como *BestComp* el mismo compuesto que OpR (o incluso WEGA), su valor Tc_S puede ser menor. Si se permite que el algoritmo se ejecute más tiempo, como con OpR, las soluciones encontradas se pueden mejorar en términos de la función objetivo. En este caso priorizamos el esfuerzo computacional. La figura 3.1 muestra un ejemplo gráfico de este hecho, específicamente la query DB09236, cuyo resultado se puede ver en la tabla 3.2. Teniendo en cuenta esta query, OpR obtiene que DB00270 es el compuesto con mayor valor de Tc_S, con un valor igual a 0.672 (ver figura 3.1a). OpF sin embargo, obtiene que el compuesto DB01115 es el más similar con un valor igual a $Tc_S = 0.615$. Finalmente, WEGA devuelve como mejor compuesto la molécula DB01433 con un valor de $Tc_S = 0.662$. Aparentemente, WEGA logra un compuesto más similar que OpF, ya que proporciona como solución un compuesto con un Tc_S más alto que el propuesta por OpF. Sin embargo, cuando OpR optimiza la query con la molécula DB01115 propuesta por OpF, proporciona un valor de Tc_S de 0.669 (ver figura 3.1b). Por el contrario, OpR da un valor de 0.662 cuando optimiza la query con el compuesto DB01433 dado por WEGA, (ver Figura 3.1c). Esto significa que la solución proporcionada por OpF es más similar en términos de forma que la de WEGA.

La tabla 3.3 muestra los tiempos entre los diferentes métodos. Claramente, el algoritmo más lento es OpR, ya que se ha ajustado para que sea robusto y preciso. Aun así, los valores de tiempo no son extremadamente altos en comparación con los otros dos métodos. Por su parte, OpF es el algoritmo más rápido, reduciendo en promedio el esfuerzo computacional de WEGA casi 3.5 veces. Además, como se puede apreciar en la columna *speedup*, cuanto menor es el número de átomos, mayor es el aumento de la velocidad obtenida por OpF. Además, es importante mencionar que OpF es capaz de adaptarse a la complejidad del problema a resolver.

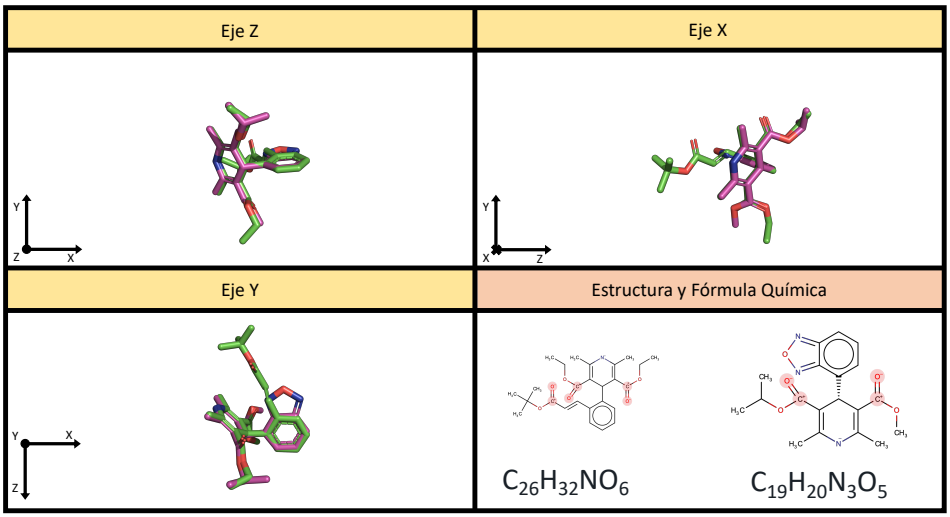
Finalmente, es interesante observar que, a pesar de la aleatoriedad incluida en algunas etapas del algoritmo OptiPharm, su variabilidad es casi insignificante, como se puede apreciar a partir de los valores de Desviación Estándar (SD, *Standard Deviation*) proporcionados en la tabla 3.3.



(a) Optimización de los compuestos DB09236 y DB00270. $Tc_S = 0.672$



(b) Optimización de los compuestos DB09236 y DB01115. $Tc_S = 0.615$



(c) Optimización de los compuestos DB09236 y DB01433.Tc=$Tc_S = 0.662$

Figura 3.1: Representación de similitud de forma entre la query DB09236 y (a) la molécula DB00270, (b) el compuesto DB01115 y (c) la molécula DB01433, cuando están optimizados usando OpR.

Tabla 3.2: Los resultados obtenidos de 40 queries de la base de datos FDA. Para cada query, se muestra su *nA* y *BestComp* con el Tc_S más alto, de acuerdo con OpR, OpF y WEGA. Tenga en cuenta que la puntuación Tc_S es igual a 1 cuando el compuesto query se compara consigo mismo en todas las instancias y algoritmos, de modo que *BestComp* realmente representa la segunda molécula más similar a la query.

query		OpR		OpF			WEGA		
nombre	*nA*	*BestComp*	Tc_S	*BestComp*	Tc_S	(i,Tc_S(OpR))	*BestComp*	Tc_S	(i,Tc_S(OpR))
DB00529	7	DB00828	0.921	DB00828	0.920	-	DB00828	0.921	-
DB00331	9	DB01189	0.940	DB01189	0.936	-	DB01189	0.940	-
DB01365	12	DB00191	0.944	DB00191	0.943	-	DB00191	0.944	-
DB01352	15	DB00306	0.891	DB00306	0.884	-	DB00237	0.872	(2, 0.872)
DB00380	19	DB00816	0.842	DB00816	0.822	-	DB00816	0.842	-
DB06216	20	DB00370	0.905	DB00370	0.902	-	DB09304	0.856	(2, 0.869)
DB00674	21	DB01619	0.865	DB01619	0.855	-	DB00370	0.850	(2, 0.850)
DB00632	23	DB00464	0.724	DB00464	0.719	-	DB00464	0.717	-
DB07615	24	DB01250	0.799	DB01250	0.797	-	DB01250	0.799	-
DB00693	25	DB01619	0.841	DB01619	0.793	-	DB01068	0.825	(2, 0.825)
DB00887	25	DB06614	0.745	DB06614	0.732	-	DB04938	0.733	(2, 0.730)
DB09219	25	DB00434	0.819	DB00792	0.805	(3, 0.812)	DB00792	0.812	(3, 0.812)
DB00351	27	DB04839	0.941	DB04839	0.936	-	DB00603	0.902	(2, 0.902)
DB00381	28	DB01023	0.819	DB01023	0.732	-	DB06712	0.707	(5, 0.706)
DB09237	28	DB01054	0.717	DB01054	0.648	-	DB01115	0.686	(4, 0.685)
DB01198	29	DB00402	0.933	DB00402	0.929	-	DB00402	0.933	-
DB00876	30	DB09039	0.664	DB05239	0.651	(3, 0.653)	DB05239	0.653	(3, 0.653)
DB01621	32	DB01148	0.694	DB01148	0.693	-	DB01148	0.694	-
DB09236	33	DB00270	0.672	DB01115	0.615	(2, 0.669)	DB01433	0.662	(3, 0.662)
DB08903	37	DB00333	0.653	DB00333	0.610	-	DB06703	0.630	(4, 0.630)
DB00728	38	DB01339	0.820	DB01339	0.816	-	DB01339	0.820	-
DB01419	42	DB06605	0.630	DB06605	0.626	-	DB06605	0.630	-
DB00320	43	DB01413	0.629	DB01413	0.618	-	DB01413	0.629	-
DB01232	49	DB01082	0.549	DB01082	0.535	-	DB01082	0.549	-
DB00246	50	DB01261	0.761	DB01261	0.738	-	DB01261	0.761	-
DB00503	50	DB00845	0.499	DB01319	0.461	(4, 0.496)	DB01319	0.498	(4, 0.496)
DB09114	50	DB08993	0.476	DB04894	0.411	(6, 0.416)	DB08993	0.477	-
DB00254	55	DB00595	0.877	DB00595	0.874	-	DB00595	0.877	-
DB00309	55	DB00541	0.634	DB00541	0.618	-	DB00541	0.634	-
DB06439	57	DB00207	0.515	DB00207	0.494	-	DB00212	0.513	(2, 0.513)
DB01196	60	DB00286	0.784	DB00286	0.779	-	DB00286	0.784	-
DB01078	66	DB00511	0.502	DB00511	0.479	-	DB00511	0.503	-
DB01590	68	DB00877	0.469	DB00385	0.459	(2, 0.464)	DB00877	0.469	-
DB04894	80	DB00364	0.482	DB00364	0.468	-	DB00864	0.453	(3, 0.453)
DB04786	86	DB01078	0.387	DB09158	0.306	(3, 0.369)	DB01078	0.387	-
DB00732	87	DB01045	0.434	DB01045	0.417	-	DB01045	0.434	-
DB00403	94	DB00035	0.394	DB06402	0.355	(4, 0.376)	DB08874	0.386	(2, 0.386)
DB00050	102	DB00569	0.396	DB00569	0.391	-	DB00569	0.396	-
DB06699	117	DB00091	0.454	DB00512	0.409	(2, 0.414)	DB09099	0.412	(3, 0.411)

Continúa en la siguiente página

query		OpR		OpF			WEGA		
nombre	*nA*	*BestComp*	Tc_S	*BestComp*	Tc_S	(i,Tc_S(OpR))	*BestComp*	Tc_S	(i,Tc_S(OpR))
DB06219	128	DB00512	0.422	DB00364	0.354	(2, 0.409)	DB00364	0.410	(2, 0.409)

Tabla 3.3: Resultados de los tiempos obtenidos por los diferentes métodos de similitud. Las columnas representan: código de DrugBank para cada molécula, su correspondiente *nA*, tiempo de ejecución promedio (en segundos) y desviación estándar obtenida por OpF y OpR (ver columnas 3 - 6), tiempo de ejecución empleado por WEGA (ver columna 7), y aceleración de OpF frente a WEGA.

query		OpF		OpR		WEGA	
nombre	*nA*	*Av*	*SD*	*Av*	*SD*	*T*	*speedup*
DB00529	7	4.8	0.008	61.2	0.560	16.4	3.4
DB00331	9	5.8	0.041	77.4	0.752	17.5	3.0
DB01365	12	7.3	0.004	96.7	0.714	16.9	2.3
DB01352	15	9.1	0.037	116.5	0.823	19.5	2.1
DB00380	19	11.0	0.028	165.1	1.425	20.4	1.9
DB06216	20	11.8	0.030	169.2	1.203	25.3	2.1
DB00674	21	12.3	0.011	169.9	1.123	20.6	1.7
DB00632	23	11.3	0.005	130.4	1.564	22.3	2.0
DB07615	24	13.4	0.010	205.4	1.385	22.4	1.7
DB00693	25	14.5	0.017	215.2	2.158	24.2	1.7
DB00887	25	14.2	0.001	213.5	1.547	21.6	1.5
DB09219	25	14.3	0.010	223.1	1.709	22.6	1.6
DB00351	27	15.3	0.017	220.7	1.980	23.4	1.5
DB00381	28	15.5	0.013	227.5	1.499	32.1	2.1
DB09237	28	15.8	0.001	227.4	1.222	22.8	1.4
DB01198	29	14.6	0.000	223.9	1.354	23.1	1.6
DB00876	30	17.1	0.002	262.0	1.878	23.7	1.4
DB01621	32	17.2	0.017	267.1	1.475	24.7	1.4
DB09236	33	18.1	0.059	280.7	2.230	27.0	1.5
DB08903	37	20.0	0.045	289.3	2.188	25.5	1.3
DB00728	38	20.3	0.032	284.6	1.787	25.8	1.3
DB01419	42	21.7	0.031	359.2	2.371	28.5	1.3
DB00320	43	22.8	0.016	355.6	2.374	25.6	1.1
DB01232	49	25.7	0.036	395.7	2.896	29.2	1.1
DB00246	50	15.5	0.073	250.7	1.719	22.6	1.5
DB00503	50	26.3	0.008	416.6	2.743	31.0	1.2
DB09114	50	23.9	0.005	388.3	2.782	31.6	1.3
DB00254	55	17.4	0.003	263.1	1.919	25.3	1.5
DB00309	55	28.5	0.022	377.3	2.626	30.8	1.1
DB06439	57	29.3	0.048	434.8	2.937	32.9	1.1
DB01196	60	15.9	0.054	244.7	1.599	27.1	1.7
DB01078	66	28.9	0.050	485.9	3.538	36.0	1.2
DB01590	68	31.9	0.010	495.1	3.297	39.4	1.2
DB04894	80	37.8	0.006	550.4	3.876	40.7	1.1
DB04786	86	32.3	0.002	598.7	4.728	45.4	1.4

Continúa en la siguiente página

query		OpF		OpR		WEGA	
nombre	*nA*	*Av*	*SD*	*Av*	*SD*	*T*	*speedup*
DB00732	87	40.1	0.015	628.5	4.147	44.2	1.1
DB00403	94	39.6	0.041	609.5	5.072	49.5	1.3
DB00050	102	45.5	0.050	664.1	4.834	51.3	1.1
DB06699	117	50.8	0.005	725.6	5.257	55.0	1.1
DB06219	128	52.0	0.090	828.4	7.030	63.4	1.2
media	46	21.7	0.024	330.0	2.408	29.7	1.6

3.5 Comparativa de OptiPharm y WEGA con las bases de datos DUD y DUD-E sin hidrógenos

En esta sección se analizará la capacidad de clasificación de los software WEGA y OptiPharm. Para ello se han calculado las métricas de los valores del Área bajo la curva ROC (Característica Operativa del Receptor) (AUC, *Area Under the Curve ROC (Receiver Operating Characteristic)*)) y el tiempo de ejecución. El valor de AUC es ampliamente utilizado en problemas de clasificación donde se utilizan experimentos a medida en los que se conoce previamente los resultados que deben obtenerse. Su valor está comprendido en el rango $[0, 1]$ donde valores por debajo de 0.5 indican una mala fiabilidad del clasificador, valores similares a 0.5 indican que la clasificación se hace aleatoriamente, y valores mayores de 0.5 determinan que la clasificación de los elementos se realiza mejor conforme más cerca es este valor a 1. Para un mayor detalle sobre su cálculo, se proporcionan en el anexo A una explicación más amplia.

Como se mencionó en la sección 3.2, para probar la confiabilidad de OptiPharm, cada instancia se ha ejecutado 100 veces y se han calculado los valores medios. Las tablas 3.4 y 3.5 muestran el rendimiento de OptiPharm (en sus dos versiones) y de WEGA cuando estos se ejecutan considerando las bases de datos DUD y DUD-E, respectivamente. Además del AUC, también se ha proporcionado la SD correspondiente. WEGA, dado que es determinista, solo se ha llevado a cabo una única ejecución para cada instancia y se han mostrado los valores correspondientes.

En términos generales, los valores de SD obtenidos para OpF y OpR son bastante pequeños, lo que indica que su variabilidad es pequeña, y que (i) convergen hacia el mismo óptimo a pesar de la aleatoriedad incluida y (ii) el tiempo de cálculo es prácticamente el mismo cuando se llevan a cabo diferentes ejecuciones de la misma instancia.

Centrándonos ahora en la tabla 3.4, es posible inferir que los tres algoritmos son equivalentes en términos de precisión de las predicciones, es decir, obtienen aproximadamente los mismos valores de AUC independientemente de la instancia considerada. De hecho, el valor medio de AUC es prácticamente igual, como se puede ver en la última fila de la tabla. Sin embargo, OpF es casi 5 veces más rápido que WEGA y más de 16 veces más rápido que OpR.

Finalmente, se pueden obtener conclusiones similares a las de la base de datos DUD-E (ver tabla 3.5). En términos de efectividad, OpR y WEGA son comparables, ya que obtienen prácticamente el mismo valor medio de AUC. Por el contrario, OpF obtiene un valor de AUC

medio ligeramente menor. Sin embargo, OpF es más de 17 veces más rápido que WEGA y más de 38 veces más rápido que OpR.

Tabla 3.4: Base de datos DUD. Para cada compuesto query, se calcularon 100 ejecuciones independientes de OpF y OpR con su valor medio de AUC y tiempo de ejecución (en segundos). Además, se proporciona también la SD para las versiones OpF y OpR. WEGA es un algoritmo determinista por lo que solo se ejecutó una vez y se incluyen su valor de AUC calculado y el tiempo de ejecución. La última fila de la tabla muestra valores medios para las queries.

	AUC					Tiempo				
	OpF		OpR		WEGA	OpF		OpR		WEGA
nombre	*Av*	*SD*	*Av*	*SD*	*AUC*	*Av*	*SD*	*Av*	*SD*	*T*
ace	0.44	0.021	0.39	0.013	0.33	15.2	0.000	278.7	0.046	31.0
ache	0.71	0.008	0.71	0.004	0.72	35.5	0.003	645.5	0.059	67.0
ada	0.71	0.011	0.67	0.003	0.66	4.9	0.000	67.8	0.011	12.5
alr2	0.28	0.012	0.24	0.003	0.22	6.8	0.000	87.3	0.015	13.9
ampc	0.75	0.020	0.70	0.005	0.71	5.0	0.000	68.6	0.013	10.9
ar	0.73	0.005	0.73	0.003	0.72	18.1	0.001	209.2	0.020	41.2
cdk2	0.58	0.010	0.60	0.010	0.59	12.4	0.000	184.3	0.026	28.7
comt	0.45	0.016	0.43	0.017	0.37	3.3	0.000	45.6	0.007	10.0
cox1	0.51	0.009	0.49	0.003	0.48	4.7	0.000	57.2	0.009	12.6
cox2	0.93	0.004	0.95	0.002	0.95	109.6	0.006	1738.5	0.112	1038.6
dhfr	0.61	0.007	0.65	0.003	0.65	83.6	0.006	1392.8	0.081	742.6
egfr	0.54	0.006	0.59	0.003	0.57	137.3	0.008	2128.5	0.100	962.1
er_agonist	0.80	0.007	0.79	0.003	0.79	17.6	0.001	228.4	0.026	120.7
er_antagonist	0.73	0.015	0.73	0.008	0.72	15.2	0.000	262.4	0.029	70.0
fgfr1	0.45	0.003	0.41	0.001	0.40	39.4	0.003	668.7	0.047	387.6
fxa	0.60	0.010	0.60	0.007	0.68	65.5	0.005	1161.2	0.073	244.6
gart	0.41	0.012	0.31	0.007	0.27	11.6	0.000	197.0	0.024	49.6
gpb	0.82	0.008	0.85	0.004	0.84	10.5	0.000	128.9	0.016	35.5
gr	0.66	0.008	0.62	0.005	0.62	27.4	0.002	365.2	0.034	53.5
hivpr	0.71	0.011	0.78	0.011	0.76	36.0	0.001	622.7	0.063	51.1
hivrt	0.75	0.010	0.75	0.011	0.75	9.8	0.000	143.8	0.019	34.0
hmga	0.75	0.015	0.75	0.012	0.77	14.9	0.000	240.7	0.027	78.5
hsp90	0.77	0.016	0.68	0.009	0.66	8.2	0.000	128.7	0.019	30.5
inha	0.53	0.009	0.61	0.007	0.60	32.4	0.002	479.7	0.045	84.4
mr	0.84	0.007	0.84	0.004	0.84	5.6	0.000	66.6	0.011	10.7
na	0.83	0.008	0.86	0.008	0.85	12.4	0.000	165.4	0.017	31.6
p38	0.45	0.012	0.50	0.003	0.47	112.7	0.006	1997.2	0.125	371.6
parp	0.46	0.008	0.50	0.003	0.49	7.8	0.000	96.3	0.016	33.8
pde5	0.74	0.009	0.75	0.008	0.75	23.5	0.001	420.6	0.038	124.9
pdgfrb	0.47	0.006	0.45	0.003	0.46	54.3	0.005	964.0	0.058	145.7
pnp	0.61	0.020	0.61	0.008	0.63	5.6	0.000	71.4	0.011	17.2
ppar_gamma	0.72	0.011	0.68	0.014	0.70	50.2	0.003	1055.6	0.086	134.2
pr	0.65	0.029	0.62	0.018	0.61	10.9	0.000	151.7	0.024	44.5
rxr_alpha	0.91	0.013	0.90	0.023	0.91	7.3	0.000	122.0	0.016	13.8
sahh	0.87	0.007	0.89	0.006	0.89	6.7	0.000	87.9	0.012	19.5
src	0.38	0.008	0.32	0.003	0.30	74.3	0.006	1388.0	0.072	272.7

Continúa en la siguiente página

	AUC					Tiempo				
	OpF		OpR		WEGA	OpF		OpR		WEGA
nombre	*Av*	*SD*	*Av*	*SD*	*AUC*	*Av*	*SD*	*Av*	*SD*	*T*
thrombin	0.57	0.013	0.50	0.009	0.55	28.6	0.001	510.2	0.045	145.4
tk	0.56	0.017	0.56	0.018	0.58	4.2	0.000	47.8	0.008	20.6
trypsin	0.33	0.009	0.28	0.006	0.26	12.6	0.000	255.4	0.024	41.0
vegfr2	0.60	0.008	0.61	0.006	0.61	21.2	0.001	323.5	0.027	49.5
media	0.63	0.011	0.62	0.007	0.61	29.1	0.002	481.4	0.038	142.2

Tabla 3.5: Base de datos DUD-E. Para cada query, se calcularon el valor medio de AUC y del tiempo de ejecución (en segundos) sobre 100 ejecuciones independientes con OpR y OpF. La SD también se proporciona para las versiones OpR y OpF. WEGA es un algoritmo determinista, por lo que solo se ejecutó una vez y se incluyen su valor de AUC y el tiempo de ejecución. La última fila de la tabla muestra valores medios para las queries.

	AUC					Tiempo				
	OpF		OpR		WEGA	OpF		OpR		WEGA
nombre	*Av*	*SD*	*Av*	*SD*	AUC	*Av*	*SD*	*Av*	*SD*	*T*
aa2ar	0.56	0.011	0.57	0.000	0.57	45.6	1.125	2656.8	24.748	1363.1
abl1	0.53	0.003	0.52	0.001	0.52	35.8	1.197	905.6	5.756	430.4
ace	0.50	0.015	0.51	0.000	0.53	27.8	0.827	1392.5	4.369	710.6
aces	0.27	0.006	0.24	0.000	0.24	26.9	0.901	1733.6	3.072	978.7
ada	0.58	0.041	0.63	0.000	0.71	5.3	0.193	245.6	1.570	232.0
ada17	0.47	0.002	0.48	0.000	0.48	47.6	1.796	1894.6	3.297	1011.2
adrb1	0.33	0.003	0.36	0.002	0.36	32.6	1.002	966.5	1.987	634.5
adrb2	0.37	0.003	0.37	0.001	0.38	19.8	0.736	1155.9	10.912	555.6
akt1	0.26	0.008	0.26	0.001	0.26	40.4	0.977	1062.5	1.992	675.6
akt2	0.34	0.004	0.41	0.001	0.39	17.0	0.563	504.1	5.271	210.9
aldr	0.50	0.007	0.54	0.001	0.54	23.0	0.789	565.1	2.701	310.2
ampc	0.52	0.007	0.63	0.000	0.64	2.5	0.096	118.9	1.736	101.8
andr	0.60	0.002	0.63	0.000	0.63	26.1	0.774	595.8	17.953	398.3
aofb	0.45	0.002	0.44	0.000	0.44	3.1	0.090	135.2	0.411	198.3
bace1	0.46	0.018	0.53	0.000	0.54	37.8	1.197	1284.8	4.041	758.4
braf	0.48	0.005	0.56	0.002	0.55	14.8	0.424	766.7	9.025	352.3
cah2	0.44	0.002	0.45	0.003	0.44	37.6	1.297	765.5	13.342	1173.8
casp3	0.44	0.001	0.41	0.000	0.39	13.8	0.384	561.4	0.763	436.6
cdk2	0.64	0.004	0.66	0.001	0.66	70.7	1.907	3007.8	71.465	837.4
comt	0.56	0.005	0.60	0.002	0.62	5.3	0.146	122.9	1.591	111.7
cp2c9	0.43	0.005	0.43	0.000	0.44	11.1	0.326	408.3	2.055	280.0
cp3a4	0.53	0.007	0.53	0.001	0.53	32.5	0.904	1370.1	6.526	430.5
csf1r	0.58	0.006	0.55	0.000	0.60	27.9	0.806	1085.7	37.667	397.4
cxcr4	0.65	0.003	0.71	0.002	0.73	4.0	0.117	231.4	1.851	112.3
def	0.55	0.008	0.69	0.000	0.69	6.5	0.188	324.2	15.676	191.6
dhi1	0.67	0.002	0.64	0.000	0.64	26.8	0.689	1200.7	7.467	703.7
dpp4	0.55	0.002	0.57	0.000	0.57	62.8	1.866	3618.4	6.684	1402.7
drd3	0.29	0.002	0.30	0.000	0.29	56.3	1.884	2085.8	6.171	1174.4
dyr	0.38	0.004	0.40	0.001	0.40	63.6	1.888	976.6	26.652	624.9

Continúa en la siguiente página

	AUC					Tiempo				
	OpF		OpR		WEGA	OpF		OpR		WEGA
nombre	*Av*	*SD*	*Av*	*SD*	AUC	*Av*	*SD*	*Av*	*SD*	*T*
egfr	0.45	0.005	0.52	0.001	0.54	54.8	1.780	3601.2	99.478	1460.8
esr1	0.64	0.003	0.64	0.000	0.63	43.2	1.385	1994.7	6.168	749.2
esr2	0.65	0.003	0.69	0.000	0.68	25.8	0.919	1300.2	2.196	693.1
fa7	0.60	0.003	0.66	0.001	0.52	111.9	1.731	2691.7	8.408	184.1
fa10	0.53	0.008	0.51	0.002	0.67	14.0	0.388	761.6	2.982	589.3
fabp4	0.62	0.010	0.69	0.009	0.67	11.1	0.425	285.8	3.176	119.0
fak1	0.67	0.002	0.69	0.002	0.67	37.6	0.863	648.8	6.135	160.3
fgfr1	0.47	0.004	0.47	0.002	0.46	1.7	0.052	50.5	0.788	50.2
fkb1a	0.67	0.008	0.68	0.001	0.72	10.4	0.308	458.1	16.792	253.5
fnta	0.47	0.004	0.55	0.000	0.55	186.2	5.634	6081.6	4.037	2102.5
fpps	0.81	0.002	0.86	0.001	0.88	8.7	0.267	250.7	1.818	221.3
gcr	0.48	0.002	0.52	0.000	0.50	29.3	0.970	1046.0	1.883	624.2
glcm	0.30	0.001	0.36	0.002	0.35	3.4	0.104	132.6	6.153	132.2
gria2	0.56	0.004	0.59	0.001	0.60	22.9	0.865	740.5	8.288	418.7
grik1	0.67	0.004	0.62	0.001	0.61	8.3	0.273	262.6	4.979	253.6
hdac2	0.31	0.002	0.34	0.000	0.35	10.6	0.354	521.0	2.440	400.5
hdac8	0.40	0.004	0.42	0.000	0.43	11.8	0.372	528.3	5.144	353.7
hivint	0.35	0.004	0.41	0.001	0.41	12.3	0.389	384.1	1.260	221.2
hivpr	0.69	0.009	0.70	0.001	0.71	133.4	3.808	4748.8	5.144	1354.1
hivrt	0.49	0.001	0.52	0.000	0.52	42.3	1.469	1107.6	4.250	573.8
hmdh	0.71	0.003	0.75	0.000	0.74	19.7	0.549	976.8	7.565	399.8
hs90a	0.60	0.008	0.63	0.001	0.64	9.7	0.189	390.3	8.946	183.7
hxk4	0.49	0.002	0.64	0.001	0.62	11.5	0.382	358.3	12.647	188.1
igf1r	0.46	0.004	0.48	0.002	0.50	31.8	1.059	1048.3	2.244	401.7
inha	0.34	0.005	0.39	0.002	0.43	2.3	0.075	130.6	0.407	79.6
ital	0.44	0.007	0.39	0.002	0.38	30.1	0.827	1157.5	2.996	459.8
jak2	0.64	0.004	0.68	0.000	0.68	8.1	0.277	412.2	4.734	283.3
kif11	0.58	0.006	0.83	0.000	0.83	8.2	0.272	606.8	6.053	318.1
kit	0.41	0.003	0.43	0.000	0.44	15.9	0.559	678.5	0.307	325.9
kith	0.65	0.002	0.69	0.003	0.70	3.7	0.145	153.3	1.227	104.5
kpcb	0.52	0.004	0.58	0.000	0.59	13.6	0.471	622.6	6.039	310.2
lck	0.43	0.002	0.46	0.001	0.44	40.6	1.237	2110.1	4.281	1121.1
lkha4	0.52	0.003	0.52	0.000	0.58	9.7	0.298	599.4	0.866	365.4
mapk2	0.61	0.003	0.65	0.000	0.65	10.4	0.316	376.4	1.190	210.0
mcr	0.59	0.002	0.64	0.000	0.63	6.1	0.200	292.4	3.096	175.4
met	0.73	0.007	0.68	0.002	0.72	46.8	1.571	2182.0	15.724	564.2
mk01	0.38	0.002	0.39	0.001	0.40	6.1	0.209	256.9	0.739	154.7
mk10	0.49	0.005	0.45	0.000	0.44	11.5	0.362	559.1	2.017	258.8
mk14	0.52	0.003	0.54	0.001	0.54	139.1	4.452	6277.2	25.295	1404.4
mmp13	0.55	0.002	0.56	0.000	0.60	63.4	2.080	3671.8	7.850	1525.6
mp2k1	0.53	0.003	0.42	0.000	0.45	11.7	0.383	722.9	17.182	339.2
nos1	0.33	0.003	0.35	0.001	0.35	6.3	0.197	366.6	8.322	267.2
nram	0.79	0.002	0.85	0.000	0.85	6.2	0.163	357.0	8.898	200.7
pa2ga	0.62	0.005	0.60	0.000	0.60	8.6	0.324	416.6	3.433	218.3

Continúa en la siguiente página

	AUC					Tiempo				
	OpF		OpR		WEGA	OpF		OpR		WEGA
nombre	*Av*	*SD*	*Av*	*SD*	AUC	*Av*	*SD*	*Av*	*SD*	*T*
parp1	0.63	0.001	0.64	0.000	0.64	43.2	1.314	1481.6	2.181	981.3
pde5a	0.56	0.002	0.59	0.000	0.56	37.4	1.222	2777.0	56.574	1243.3
pgh1	0.72	0.004	0.70	0.000	0.71	15.3	0.412	620.4	1.826	425.2
pgh2	0.74	0.001	0.79	0.000	0.79	35.5	1.161	1130.5	2.544	791.9
plk1	0.47	0.006	0.53	0.000	0.54	11.3	0.346	797.5	6.744	267.3
pnph	0.70	0.001	0.74	0.000	0.74	5.9	0.185	264.4	1.521	212.8
ppara	0.75	0.003	0.76	0.000	0.77	39.3	1.456	2109.3	27.199	870.2
ppard	0.34	0.002	0.47	0.001	0.44	31.7	0.856	1557.1	2.223	503.6
pparg	0.43	0.002	0.45	0.001	0.45	69.3	1.942	2867.4	34.034	1122.2
prgr	0.69	0.002	0.72	0.001	0.71	33.8	1.017	1148.7	12.219	469.9
ptn1	0.29	0.005	0.31	0.001	0.30	8.9	0.264	348.1	1.425	290.6
pur2	0.26	0.009	0.37	0.000	0.33	4.9	0.153	242.8	1.844	146.9
pygm	0.62	0.005	0.58	0.000	0.57	5.9	0.162	241.4	1.812	173.2
pyrd	0.80	0.001	0.84	0.000	0.85	8.2	0.237	343.0	0.970	233.1
reni	0.56	0.003	0.59	0.002	0.58	39.6	1.241	970.5	21.357	292.4
rock1	0.52	0.002	0.55	0.000	0.54	4.3	0.167	216.7	3.338	207.4
rxra	0.49	0.003	0.61	0.000	0.60	8.5	0.312	410.0	10.792	258.6
sahh	0.60	0.003	0.87	0.000	0.86	2.1	0.105	123.9	0.394	131.8
src	0.53	0.002	0.55	0.002	0.60	271.2	7.781	4995.2	6.656	1318.6
tgfr1	0.49	0.003	0.60	0.001	0.59	10.5	0.373	514.3	9.723	350.7
thb	0.75	0.001	0.79	0.000	0.81	12.8	0.428	651.8	1.963	321.2
thrb	0.43	0.003	0.45	0.000	0.45	71.5	2.444	2427.1	114.339	1205.4
try1	0.56	0.001	0.57	0.000	0.57	60.8	2.151	2483.4	50.893	1123.2
tryb1	0.36	0.003	0.38	0.000	0.39	8.3	0.275	555.0	2.471	277.8
tysy	0.61	0.007	0.65	0.002	0.66	16.4	0.525	705.4	0.347	266.7
urok	0.40	0.002	0.40	0.000	0.41	13.8	0.466	511.1	1.834	342.5
vgfr2	0.60	0.003	0.57	0.000	0.59	42.6	1.154	1816.6	16.649	902.3
wee1	0.47	0.018	0.65	0.001	0.62	12.1	0.377	695.5	5.638	204.3
xiap	0.76	0.010	0.79	0.004	0.78	16.1	0.448	530.0	6.233	187.4
media	0.53	0.005	0.56	0.001	0.56	30.1	0.912	1152.9	10.256	516.6

3.6 Análisis de las predicciones cuando se consideran o no hidrógenos en los compuestos.

WEGA no considera los átomos de hidrógeno durante la optimización siendo esta una práctica común para la mayoría de las herramientas en el escenario actual, ya que la evaluación sin hidrógenos requiere menos tiempo. Sin embargo, esta simplificación puede tener graves consecuencias en el proceso de LBVS. En esta sección se analiza el efecto de la exclusión de los hidrógenos de las moléculas durante la optimización de la similitud de forma. Para ello se utilizará OptiPharm, que sí puede ser configurado para que incluya los hidrógenos en las comparaciones entre compuestos.

La tabla 3.6 muestra el número de átomos para las 40 queries seleccionadas de la base de datos

de la FDA cuando se consideran los hidrógenos y cuando no (columnas 2 y 6 respectivamente). Además, se muestra la molécula *BestComp* del conjunto de datos de la FDA, que obtiene el mayor valor de Tc_S cuando las queries incluyen los hidrógenos y cuando no. Hay que tener en cuenta que estos experimentos se realizaron utilizando OpR ya que, de acuerdo con los resultados anteriores, es el algoritmo que mejores resultados ofrece. Para reflejar de manera más exhaustiva los resultados, también se ha incluido el tiempo medio de ejecución (en segundos), en ambos casos. Como se puede ver, en 15 de los 40 casos, la molécula *BestComp* difiere, dependiendo de si los hidrógenos se consideran o no. Además, y como se esperaba, el tiempo de ejecución disminuye cuando no se consideran los hidrógenos (ver columnas 5 y 9). Esto significa que excluir los hidrógenos de las moléculas no es una simplificación apropiada. Aunque las ejecuciones pueden realizarse más rápido, la molécula con mayor similitud de forma puede cambiar.

Finalmente, para una comparación justa en términos de valor de Tc_S, el *BestComp* optimizado obtenido por OpR cuando no se consideran hidrógenos se ha vuelto a evaluar, pero considerando ahora los hidrógenos. Como podemos ver, el valor del la función objetivo obtenido es siempre menor que el obtenido cuando se incluyen los hidrógenos (comparando las columnas 8 y 10). Esto significa que la molécula *BestComp* encontrada por OpR cuando se consideran los hidrógenos es de hecho más similar a la propuesta cuando se excluyen los hidrógenos. La figura 3.2 ilustra este hecho.

Tabla 3.6: Los resultados obtenidos por OpR para 40 queries de la base de datos FDA. Se llevaron a cabo dos experimentos, uno excluyendo los átomos de hidrógeno para todas las moléculas (una práctica común en la mayoría de las herramientas de LBVS en la literatura) y el otro considerando los hidrógenos en todas las moléculas. Para cada estudio y query, se muestra su *nA* con y sin hidrógenos, el *BestComp* con el mayor Tc_S y el tiempo de ejecución, en segundo lugar. Finalmente, el *BestComp* optimizado obtenido cuando no se consideran hidrógenos se vuelve a evaluar, pero incluye los hidrógenos (última columna). Se han sombreado las filas donde el compuesto *BestComp* obtenido con y sin hidrógenos es distinto.

	Con hidrógenos				Sin hidrógenos				*BestComp* sin H evaluado con H
query	*nA*	*BestComp*	Tc_S	T	*nA*	*BestComp*	Tc_S	T	Tc_S
DB00529	10	DB09294	0.869	135.5	7	DB00828	0.921	61.2	0.701
DB00331	20	DB09210	0.862	255.8	9	DB01189	0.940	77.4	0.710
DB01352	29	DB00306	0.889	361.1	15	DB00306	0.891	116.5	0.884
DB01365	30	DB00191	0.935	406.9	12	DB00191	0.944	96.7	0.928
DB00380	35	DB01041	0.852	477.4	19	DB00816	0.842	165.1	0.802
DB06216	37	DB00370	0.876	500.1	20	DB00370	0.905	169.2	0.874
DB00693	37	DB01619	0.863	553.4	25	DB01619	0.841	215.2	0.854
DB07615	40	DB00721	0.790	576.5	24	DB01250	0.799	205.4	0.713
DB09219	40	DB01320	0.845	636.2	25	DB00434	0.819	223.1	0.764
DB00674	42	DB01619	0.801	556.7	21	DB01619	0.865	169.9	0.786
DB01198	45	DB00402	0.892	624.7	27	DB00402	0.933	223.9	0.890
DB00887	45	DB00837	0.742	613.0	25	DB06614	0.745	213.5	0.686
DB00246	50	DB01261	0.756	737.6	28	DB01261	0.761	250.7	0.751
DB00381	53	DB01023	0.828	728.6	28	DB01023	0.819	227.5	0.823
DB09237	54	DB01054	0.752	759.4	28	DB01054	0.717	227.4	0.745

Continúa en la siguiente página

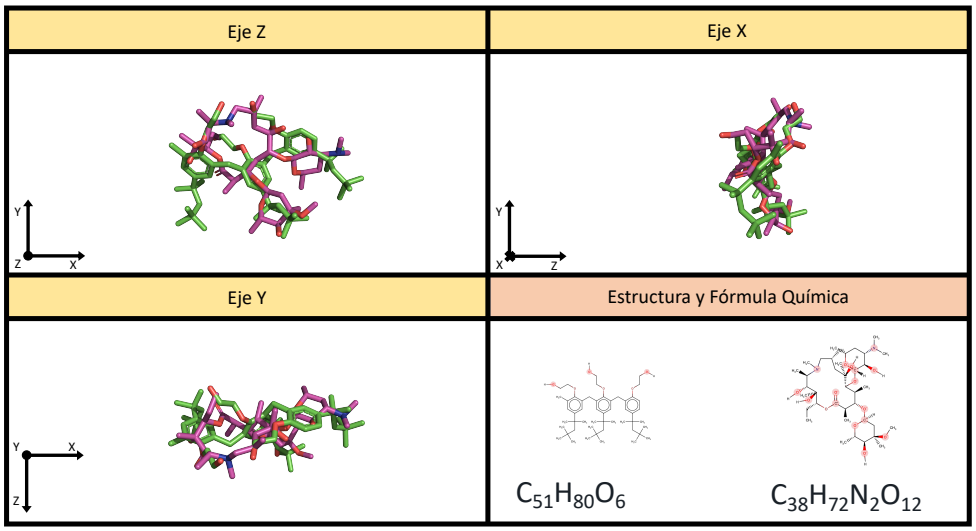
	Con hidrógenos				Sin hidrógenos				*BestComp* sin H evaluado con H
nombre	*nA*	*BestComp*	Tc_S	*T*	*nA*	*BestComp*	Tc_S	*T*	Tc_S
DB00876	54	DB09039	0.674	800.8	30	DB09039	0.664	262.0	0.665
DB00254	55	DB00595	0.848	814.7	32	DB00595	0.877	263.1	0.838
DB00351	57	DB04839	0.934	748.3	27	DB04839	0.941	220.7	0.928
DB01196	60	DB00286	0.797	820.4	29	DB00286	0.784	244.7	0.794
DB01621	66	DB01148	0.715	924.2	33	DB01148	0.694	267.1	0.708
DB09236	66	DB01054	0.682	940.2	32	DB00270	0.672	280.7	0.615
DB08903	69	DB00333	0.679	968.5	37	DB00333	0.653	289.3	0.673
DB00632	69	DB00464	0.740	696.4	23	DB00464	0.724	130.4	0.732
DB01419	70	DB06605	0.671	1086.4	42	DB06605	0.630	359.2	0.667
DB00320	80	DB00728	0.617	1139.0	43	DB01413	0.629	355.6	0.596
DB00728	91	DB01339	0.839	1094.0	38	DB01339	0.820	284.6	0.837
DB00503	98	DB00701	0.541	1465.3	50	DB00845	0.499	416.6	0.442
DB01232	100	DB00212	0.617	1411.8	49	DB01082	0.549	395.7	0.581
DB00309	110	DB00541	0.624	1348.2	55	DB00541	0.634	377.3	0.621
DB04786	120	DB00511	0.432	1657.8	86	DB01078	0.387	598.7	0.405
DB09114	130	DB08993	0.512	1799.6	50	DB08993	0.476	388.3	0.510
DB06439	137	DB00207	0.591	1871.5	57	DB00207	0.515	434.8	0.533
DB01078	140	DB00511	0.582	1819.4	66	DB00511	0.502	485.9	0.570
DB01590	151	DB00877	0.557	1995.5	68	DB00877	0.469	495.1	0.545
DB04894	152	DB00646	0.537	1797.1	80	DB00364	0.482	550.4	0.495
DB00403	167	DB08874	0.470	2130.0	94	DB00035	0.394	609.5	0.446
DB00732	169	DB06287	0.484	2204.4	87	DB01045	0.434	628.5	0.470
DB00050	194	DB00569	0.489	2248.0	102	DB00569	0.396	664.1	0.483
DB06699	221	DB09099	0.514	2482.6	117	DB00091	0.454	725.6	0.496
DB06219	229	DB00512	0.443	2796.0	128	DB00512	0.422	828.4	0.414
media	86		0.704	1124.6	44		0.686	330.0	0.674

Además, el impacto en la clasificación cuando se consideran los átomos de hidrógeno también se ha evaluado utilizando las bases de datos DUD y DUD-E. Se ha utilizado en la comparación tanto los algoritmos OpF y OpR. Los resultados correspondientes se muestran en las tablas 3.7 y 3.8. WEGA no se ha incluido en el estudio ya que nunca considera los hidrógenos.

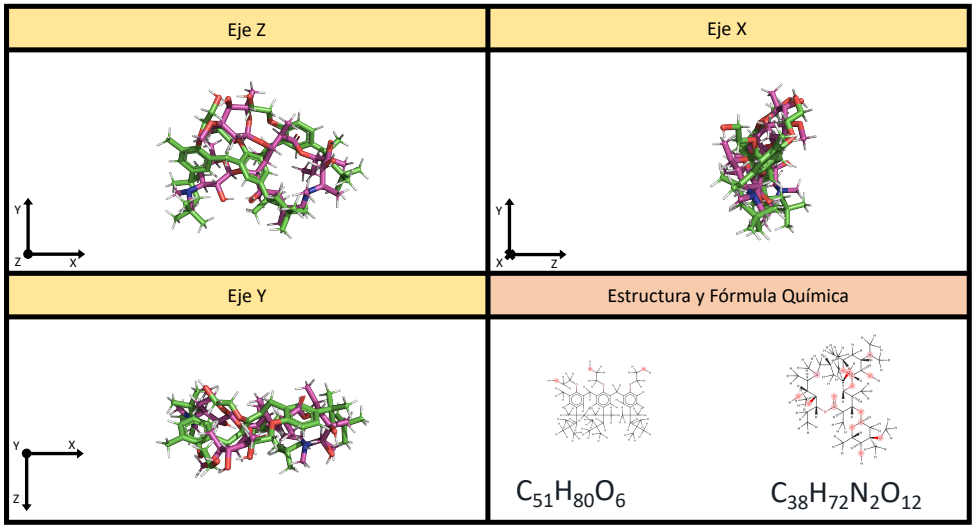
En términos generales, el valor medio de AUC aumenta ligeramente cuando se consideran los átomos de hidrógeno en la base de datos DUD, tanto para OpF como para OpR como se puede ver en la última fila de las tablas 3.4 y 3.7. En particular, se ha obtenido un incremento de 0.03 (resp. 0.01) para OpR (resp. OpF). Además, para 23 de 40 casos, OpR obtiene mejores valores de AUC cuando se consideran los hidrógenos. En cuanto a OpF, ocurre en 20 de 40 instancias.

La misma tendencia creciente se puede apreciar en el valor medio de AUC cuando se considera la base de datos DUD-E (ver tablas 3.5 y 3.8). En este caso, se ha obtenido un incremento de 0.02 para los algoritmos OpR y OpF. Tanto OpR como OpF obtienen mejores valores de AUC en más de la mitad de los casos (58 de 102 para OpR y 67 de 102 para OpF).

En términos generales, considerar los hidrógenos aumenta el tiempo medio de computación



(a) Compuestos sin hidrógenos. Tc=0.515



(b) Compuestos con hidrógenos. Tc=0.591

Figura 3.2: El compuesto query DB06439 está representado por la estructura verde. El compuesto BestComp DB00207 es el esqueleto de color rosa. Los hidrógenos se representan mediante palillos blancos. Los colores permanecen fijos. (a) Compuestos sin hidrógenos. $Tc_S = 0.515$. (b) Compuestos con hidrógenos $Tc_S = 0.591$.

como se puede ver si comparamos nuevamente las tablas 3.4 y 3.7 para la base de datos DUD, y las tablas 3.5 y 3.8 para la DUD-E. El tiempo aumenta 2.9 veces para OpR y OpF cuando se evalúa la base de datos DUD. Para el caso de DUD-E, el aumento es de 4.8x y 5.7x para OpR y OpF, respectivamente. Por supuesto, cuanto mayor sea el número de átomos considerados para un compuesto, mayor será el tiempo de cálculo asociado a su evaluación, pero más realista será el valor de la función objetivo asociada.

Tabla 3.7: Base de datos DUD con hidrógenos. Para cada compuesto query, se calculó el valor medio de AUC y el tiempo medio de ejecución (en segundos) de 100 ejecuciones independientes con OpF y OpR. Para completar la información, la SD también se proporciona para las versiones OpF y OpR. En ningún caso este valor es 0, por lo que se ha indicado con 0.000. La última fila de la tabla muestra los valores medios de las moléculas queries.

	AUC				Tiempo			
	OpF		OpR		OpF		OpR	
nombre	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*
ace	0.42	0.021	0.40	0.001	51.3	0.343	894.3	8.584
ache	0.68	0.007	0.72	0.002	132.8	0.336	2448.1	95.702
ada	0.75	0.021	0.79	0.006	15.3	0.092	227.9	10.047
alr2	0.48	0.009	0.46	0.007	15.0	0.073	187.4	6.656
ampc	0.73	0.015	0.74	0.013	9.2	0.021	131.4	1.561
ar	0.84	0.003	0.86	0.003	66.5	0.151	748.1	37.040
cdk2	0.60	0.011	0.62	0.003	30.1	0.143	449.6	11.774
comt	0.41	0.008	0.40	0.008	8.7	0.041	136.0	7.498
cox1	0.58	0.006	0.59	0.001	12.1	0.035	141.6	9.842
cox2	0.88	0.005	0.90	0.001	237.3	0.578	3768.5	99.262
dhfr	0.53	0.007	0.59	0.004	217.4	0.432	3946.1	77.654
egfr	0.57	0.004	0.56	0.002	379.9	0.484	5896.4	131.069
er_agonist	0.71	0.010	0.74	0.003	59.1	0.324	751.7	20.644
er_antagonist	0.73	0.008	0.69	0.004	52.9	0.209	887.3	23.421
fgfr1	0.46	0.002	0.42	0.000	112.2	0.309	1782.4	31.747
fxa	0.61	0.011	0.66	0.009	166.9	0.443	3089.2	41.870
gart	0.34	0.013	0.28	0.011	28.7	0.183	469.3	5.374
gpb	0.82	0.008	0.85	0.002	27.9	0.178	329.8	3.580
gr	0.76	0.011	0.77	0.004	95.1	0.280	1222.8	64.358
hivpr	0.74	0.007	0.74	0.010	113.9	0.732	2049.9	94.105
hivrt	0.69	0.009	0.70	0.008	31.1	0.174	470.9	17.540
hmga	0.82	0.008	0.84	0.004	56.6	0.162	855.8	23.483
hsp90	0.81	0.015	0.77	0.012	26.2	0.063	412.6	18.489
inha	0.53	0.005	0.59	0.010	89.5	0.289	1392.1	43.314
mr	0.86	0.004	0.87	0.003	21.4	0.092	235.4	6.255
na	0.80	0.009	0.83	0.002	40.0	0.275	479.8	9.484
p38	0.37	0.006	0.31	0.004	346.9	0.598	6491.8	129.148
parp	0.59	0.006	0.59	0.004	19.2	0.126	232.4	8.260
pde5	0.75	0.006	0.77	0.006	78.8	0.473	1399.2	12.286
pdgfrb	0.46	0.008	0.44	0.004	143.2	0.893	2704.0	93.157
pnp	0.68	0.017	0.71	0.004	14.9	0.054	193.9	1.978
ppar_gamma	0.73	0.012	0.73	0.006	139.5	0.172	3000.9	40.167
pr	0.66	0.013	0.68	0.011	36.8	0.274	544.4	25.760
rxr_alpha	0.87	0.015	0.89	0.023	25.5	0.152	414.1	9.421
sahh	0.81	0.012	0.88	0.006	15.5	0.036	227.1	13.657
src	0.46	0.005	0.44	0.002	219.2	0.510	3727.8	73.533
thrombin	0.57	0.006	0.56	0.010	92.9	0.210	1517.8	16.977
tk	0.64	0.011	0.65	0.003	11.6	0.065	125.6	3.786

Continúa en la siguiente página

	AUC				Tiempo			
	OpF		OpR		OpF		OpR	
nombre	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*
trypsin	0.30	0.008	0.27	0.004	36.3	0.187	733.3	10.189
vegfr2	0.60	0.007	0.62	0.003	54.0	0.280	861.6	53.930
media	0.64	0.009	0.65	0.005	83.3	0.262	1389.5	34.815

Tabla 3.8: Base de datos DUD-E con hidrógenos. Para cada compuesto query, se calculó el valor medio de AUC y el tiempo de ejecución medio (en segundos) de 100 ejecuciones independientes con OpR y OpF. Además, se proporciona la SD también para las versiones OpR y OpF. En ningún caso este valor es 0, por lo que se ha indicado con 0.000. La última fila de la tabla muestra valores medios para las moléculas queries.

	AUC				Tiempo			
	OpF		OpR		OpF		OpR	
nombre	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*
aa2ar	0.54	0.001	0.54	0.000	256.3	15.165	12648.0	307.385
abl1	0.58	0.003	0.56	0.000	184.4	3.043	4178.4	107.358
ace	0.63	0.001	0.63	0.000	174.3	6.823	6514.7	169.817
aces	0.23	0.001	0.22	0.000	194.1	7.856	10542.9	234.978
ada	0.70	0.003	0.68	0.000	40.2	1.630	1435.5	50.096
ada17	0.57	0.001	0.53	0.000	235.6	1.251	9711.7	254.865
adrb1	0.39	0.002	0.41	0.001	238.1	4.965	5819.1	180.508
adrb2	0.42	0.001	0.41	0.000	167.7	9.623	8295.1	243.728
akt1	0.29	0.003	0.26	0.001	205.4	3.847	5113.7	105.351
akt2	0.43	0.002	0.47	0.000	115.8	2.685	2762.0	78.460
aldr	0.55	0.006	0.56	0.001	87.3	0.626	2156.1	84.687
ampc	0.56	0.015	0.68	0.000	12.5	0.532	465.2	11.388
andr	0.75	0.001	0.78	0.000	196.9	6.801	3845.4	93.711
aofb	0.41	0.001	0.41	0.000	24.1	1.065	941.4	14.389
bace1	0.53	0.003	0.58	0.000	303.5	7.397	8931.4	248.917
braf	0.52	0.003	0.53	0.000	102.8	3.975	4113.9	105.411
cah2	0.51	0.001	0.50	0.002	145.9	4.771	2636.3	63.463
casp3	0.48	0.001	0.45	0.000	88.8	3.199	2751.9	63.384
cdk2	0.63	0.002	0.64	0.000	407.1	8.501	14337.1	270.933
comt	0.56	0.003	0.63	0.005	23.5	1.007	441.1	12.081
cp2c9	0.45	0.002	0.45	0.000	68.4	2.496	1980.4	33.515
cp3a4	0.54	0.004	0.55	0.000	211.5	6.706	7613.5	271.865
csf1r	0.54	0.001	0.51	0.000	146.7	0.106	5659.4	189.853
cxcr4	0.69	0.001	0.75	0.000	30.8	0.612	1712.4	66.771
def	0.72	0.002	0.76	0.000	55.5	1.596	2013.0	40.488
dhi1	0.75	0.002	0.75	0.000	189.5	4.195	6446.4	158.161
dpp4	0.61	0.001	0.62	0.000	328.8	14.946	15566.7	374.754
drd3	0.39	0.001	0.37	0.000	431.7	13.124	14175.3	269.919
dyr	0.38	0.002	0.42	0.003	373.8	4.095	5729.7	96.321
egfr	0.51	0.002	0.50	0.000	336.4	4.806	18151.4	354.857
esr1	0.60	0.002	0.57	0.001	240.5	0.841	10530.6	293.861

Continúa en la siguiente página

	AUC				Tiempo			
	OpF		OpR		OpF		OpR	
nombre	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*
esr2	0.63	0.003	0.64	0.000	207.5	5.593	8166.9	185.100
fa10	0.61	0.001	0.63	0.004	628.2	10.031	13762.4	325.381
fa7	0.50	0.006	0.48	0.001	88.6	2.803	4005.9	117.438
fabp4	0.67	0.005	0.74	0.003	51.3	0.834	1366.2	49.416
fak1	0.60	0.006	0.71	0.001	163.6	3.498	2801.9	81.060
fgfr1	0.47	0.001	0.47	0.002	9.5	0.613	281.7	5.810
fkb1a	0.73	0.005	0.78	0.001	72.1	3.259	2286.2	71.672
fnta	0.48	0.001	0.54	0.001	1131.1	13.666	33347.0	569.040
fpps	0.75	0.001	0.78	0.000	37.0	2.852	902.0	23.601
gcr	0.62	0.001	0.64	0.000	194.4	5.002	5936.8	117.042
glcm	0.28	0.001	0.33	0.003	21.8	0.044	923.6	32.983
gria2	0.55	0.002	0.58	0.000	100.3	6.522	3159.4	95.870
grik1	0.57	0.004	0.54	0.000	45.2	4.290	1198.1	22.338
hdac2	0.36	0.003	0.39	0.000	55.1	1.536	2752.9	88.176
hdac8	0.36	0.004	0.40	0.000	83.5	2.987	3001.7	74.002
hivint	0.38	0.001	0.38	0.000	63.8	2.193	1542.3	51.847
hivpr	0.73	0.001	0.76	0.000	764.7	0.995	26933.4	678.027
hivrt	0.52	0.001	0.56	0.001	233.7	9.952	5961.3	173.759
hmdh	0.80	0.004	0.85	0.000	127.5	1.861	4998.3	136.453
hs90a	0.65	0.002	0.66	0.000	56.4	1.329	1772.9	26.219
hxk4	0.50	0.003	0.65	0.000	59.9	2.133	1488.3	41.273
igf1r	0.43	0.004	0.46	0.001	174.9	5.849	5161.5	144.369
inha	0.42	0.004	0.40	0.000	11.7	0.104	680.8	28.818
ital	0.44	0.003	0.41	0.002	129.6	0.267	5063.5	158.899
jak2	0.69	0.002	0.72	0.000	48.4	3.409	2058.1	58.984
kif11	0.68	0.003	0.83	0.000	54.4	3.115	3439.9	123.658
kit	0.38	0.001	0.38	0.000	91.8	8.589	3238.8	110.014
kith	0.69	0.003	0.72	0.001	24.1	0.997	766.2	9.368
kpcb	0.53	0.005	0.57	0.000	94.7	2.935	3258.8	90.824
lck	0.40	0.001	0.41	0.000	247.0	12.443	11895.8	307.804
lkha4	0.57	0.003	0.57	0.000	51.1	0.528	3373.8	105.163
mapk2	0.61	0.001	0.63	0.001	60.5	3.590	1820.0	67.425
mcr	0.73	0.001	0.78	0.000	46.3	2.394	1616.9	48.168
met	0.68	0.005	0.71	0.005	250.3	6.843	11546.8	450.466
mk01	0.39	0.002	0.44	0.000	39.0	1.265	1259.4	33.755
mk10	0.46	0.002	0.45	0.000	70.2	2.139	2520.8	86.614
mk14	0.53	0.002	0.54	0.001	692.5	14.399	28472.3	565.522
mmp13	0.58	0.000	0.61	0.000	358.3	18.123	18288.5	482.792
mp2k1	0.54	0.001	0.45	0.000	69.3	6.095	3429.1	71.075
nos1	0.35	0.001	0.34	0.000	58.7	2.016	2571.5	73.863
nram	0.86	0.002	0.88	0.000	43.6	3.260	1839.6	46.528
pa2ga	0.66	0.004	0.67	0.000	59.1	5.073	2310.8	67.884
parp1	0.66	0.000	0.64	0.000	261.3	7.200	7534.6	164.474
pde5a	0.48	0.003	0.50	0.000	127.7	0.662	7966.8	132.170

Continúa en la siguiente página

	AUC				Tiempo			
	OpF		OpR		OpF		OpR	
nombre	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*	*Av*	*SD*
pgh1	0.70	0.003	0.70	0.000	102.0	5.014	3045.1	89.026
pgh2	0.70	0.002	0.72	0.000	201.6	7.323	4841.0	142.901
plk1	0.54	0.003	0.60	0.000	59.5	0.316	4137.3	131.707
pnph	0.67	0.005	0.72	0.000	29.9	0.095	1321.8	49.860
ppara	0.65	0.002	0.67	0.000	184.1	2.299	9214.1	279.082
ppard	0.37	0.006	0.39	0.001	194.5	4.637	7555.7	234.371
pparg	0.37	0.001	0.41	0.000	388.7	12.057	13606.9	308.089
prgr	0.75	0.004	0.80	0.000	208.6	4.872	5894.0	155.765
ptn1	0.36	0.002	0.35	0.000	36.5	2.168	1233.6	29.355
pur2	0.37	0.007	0.47	0.000	28.7	1.284	1035.2	31.177
pygm	0.62	0.002	0.61	0.000	36.0	1.144	1091.7	30.718
pyrd	0.81	0.004	0.81	0.000	34.4	0.277	1474.6	37.570
reni	0.65	0.005	0.68	0.001	253.1	3.662	6085.1	234.368
rock1	0.56	0.001	0.56	0.000	26.4	0.190	1414.9	44.631
rxra	0.55	0.006	0.72	0.000	50.5	0.268	2236.7	58.555
sahh	0.61	0.002	0.85	0.000	9.6	0.279	549.8	12.608
src	0.55	0.001	0.57	0.003	1187.2	15.975	23435.6	686.152
tgfr1	0.53	0.001	0.53	0.000	48.8	0.365	2329.2	76.982
thb	0.75	0.003	0.79	0.000	83.2	3.025	3150.5	82.192
thrb	0.48	0.002	0.50	0.000	444.4	3.482	13973.6	228.488
try1	0.60	0.002	0.59	0.000	384.5	0.510	12992.8	261.488
tryb1	0.38	0.004	0.42	0.000	63.8	2.376	3221.7	92.226
tysy	0.58	0.004	0.60	0.000	72.7	0.099	3038.2	113.099
urok	0.37	0.001	0.38	0.000	77.0	0.357	2944.9	92.744
vgfr2	0.54	0.001	0.52	0.000	242.1	5.486	9023.2	150.967
wee1	0.56	0.002	0.70	0.001	77.2	2.831	3547.3	120.772
xiap	0.80	0.005	0.86	0.000	101.8	4.684	3272.1	112.115
media	0.55	0.003	0.58	0.000	171.6	4.183	5878.3	148.367

3.7 Conclusiones

En este capítulo se ha mostrado el rendimiento de OptiPharm en términos de precisión de la predicción y tiempo de ejecución al procesar bases de datos de referencia utilizadas en la literatura. La comparación realizada con WEGA muestra que OptiPharm ofrece la misma precisión predictiva pero a un costo computacional mucho menor obteniendo 5 veces más rápido los resultados. Otra de las ventajas del método en comparación con WEGA es que su algoritmo de optimización es fácilmente parametrizable y, por lo tanto, se adapta a las base de datos dada dependiendo del tamaño molecular medio. Además, hay que tener en cuenta que OptiPharm, a diferencia de WEGA, permite la optimización incluyendo los átomos de hidrógeno de los compuestos. Los resultados han demostrado que considerarlos mejora las predicciones, aunque es más costoso desde el punto de vista computacional.

4. LBVS basado en la similitud del potencial electrostático

En este capítulo se analiza la eficiencia y eficacia de OptiPharm para abordar el problema de LBVS basado en la similitud del potencial electrostático. En la sección 4.1 se define el problema de optimización y se resumen los enfoques previos que se han publicado en la literatura para resolverlo. En la sección 4.2 se indican las condiciones bajo las que se realizan los experimentos. Las secciones 4.3 y 4.4 muestran los resultados obtenidos y las comparaciones entre la metodología tradicional y la nueva que se propone en esta tesis. Finalmente, la sección 4.5 recoge las conclusiones de este capítulo.

4.1 Problema de optimización

En problemas de LBVS es una práctica común y ampliamente extendida el utilizar el propio descriptor que se optimiza para seleccionar los mejores compuestos desde una base de datos [29, 107, 120]. Sin embargo, el LBVS basado en la similitud del potencial electrostático difiere de esta metodología. Las predicciones presentes en la literatura no se basan únicamente en el propio descriptor sino que se utiliza también la similitud de forma para ello [121-124].

En términos generales, los trabajos de la literatura siguen una misma metodología, la cual hemos autodenominado LBVS-Shape, aunque pueden diferir en el procedimiento de selección utilizado para determinar los compuestos propuestos como mejores predicciones [125-133]. Esencialmente, el método LBVS-Shape basa sus predicciones en un proceso de prefiltrado que consiste en identificar los H compuestos candidatos de la base de datos con la mayor similitud de forma. Después de eso, para cada compuesto seleccionado, se calcula la similitud electrostática en la superposición óptima obtenida en la etapa anterior. Finalmente, la molécula con el mayor valor de similitud de potencial electrostático se selecciona como la solución final. El valor de H no es fijo, ya que depende del caso de estudio en particular. Por lo general, H suele ser menor del 10 % del total de compuestos en la base de datos [124, 129, 130].

En esta tesis se considera que la búsqueda de los mejores compuestos basados en el prefiltrado basado en la similitud de forma puede ser contraproducente, ya que de un lado, la selección de un valor bajo de H puede descartar muchos compuestos prometedores, lo que puede tener un impacto significativo en los resultados finales y de otro, el problema de optimización puede converger a soluciones subóptimas. Así pues, el nuevo enfoque que se propone, definido matemáticamente en el problema 4.1 y que se ha llamado LBVS-Electrostatic se basa en la idea de que la función objetivo utilizada para guiar el método de optimización debe basarse principalmente en la similitud electrostática.

Problema 4.1

$$P_{Eqt} = \begin{cases} \text{máx} & Tc_{Eqt}(\Theta_t, c1_t, c2_t, \Delta_t), \\ \text{s.t.} & \Theta_t \in [0, 2\pi](\text{rad}), \\ & c1_t, c2_t \in [(\underline{-box_{t_x}, -box_{t_y}, -box_{t_z}}), (\overline{box_{t_x}, box_{t_y}, box_{t_z}})](\text{Å}), \\ & \Delta_t \in [(\underline{-\Delta_{qt_x}, -\Delta_{qt_y}, -\Delta_{qt_z}}), (\overline{\Delta_{qt_x}, \Delta_{qt_y}, \Delta_{qt_z}})](\text{Å}) \end{cases} \tag{4.1}$$

P_{Eqt} se puede entender como un problema de maximización en el que dados un compuesto query q y una base de datos de compuestos DB, encontrar para cada compuesto $t \in DB$ el máximo valor de similitud para cada par $q,t(Tc_{Eqt})$ optimizando las variables de decisión dentro de sus límites ($\Theta, c1, c2$ y Δ) definidas en el capítulo 2. Estos límites se calculan dinámicamente para cada pareja de compuestos y dependen de los tamaños de estos últimos.

Una vez se conocen los valores de similitud para cada t respecto de q, se selecciona aquel que tenga el valor más alto:

$$BestComp = \text{máx}(P_{Eqt}) \; \forall t \in DB$$

4.2 Configuración de los estudios computacionales

En esta tesis, la metodología LBVS-Shape se ha ejecutado considerando ROCS como algoritmo de optimización de la similitud de superposición de volumen mediante funciones gaussianas [25, 28] cuyo valor de similitud es Tc_R y está definido en la ecuación 1.6. En relación con la similitud electrostática se va a calcular con el kit de herramientas ZAP (ver ecuación 1.9). Este software se ha descargado sin modificaciones desde el sitio web original [134]. Cabe destacar que ROCS y ZAP son las herramientas más ampliamente utilizadas en la literatura para LBVS basadas en la forma y la similitud electrostática [135-139].

En lo referente al método LBVS-Electrostatic, se ha utilizado OptiPharm [107] como algoritmo para probar nuestra hipótesis. La similitud electrostática entre dos compuestos también se ha calculado utilizando el kit de herramientas de ZAP [134]. Este enfoque asegura que las comparaciones entre metodologías se realicen en las mismas condiciones, garantizando un estudio justo y completo mediante una comparación con los métodos más modernos. Dado que OptiPharm es un algoritmo parametrizable, se ha utilizado la configuración OpR ya empleada con buenos resultados en el capítulo 3. De forma resumida, los parámetros de esta configuración son: $N = 200000$ evaluaciones, $M = 5$ soluciones iniciales, $t_{max} = 5$ iteraciones y $R_{t_{max}} = 1$ como el radio de las soluciones del último nivel. Además, al ser heurístico, se ha ejecutado 50 veces

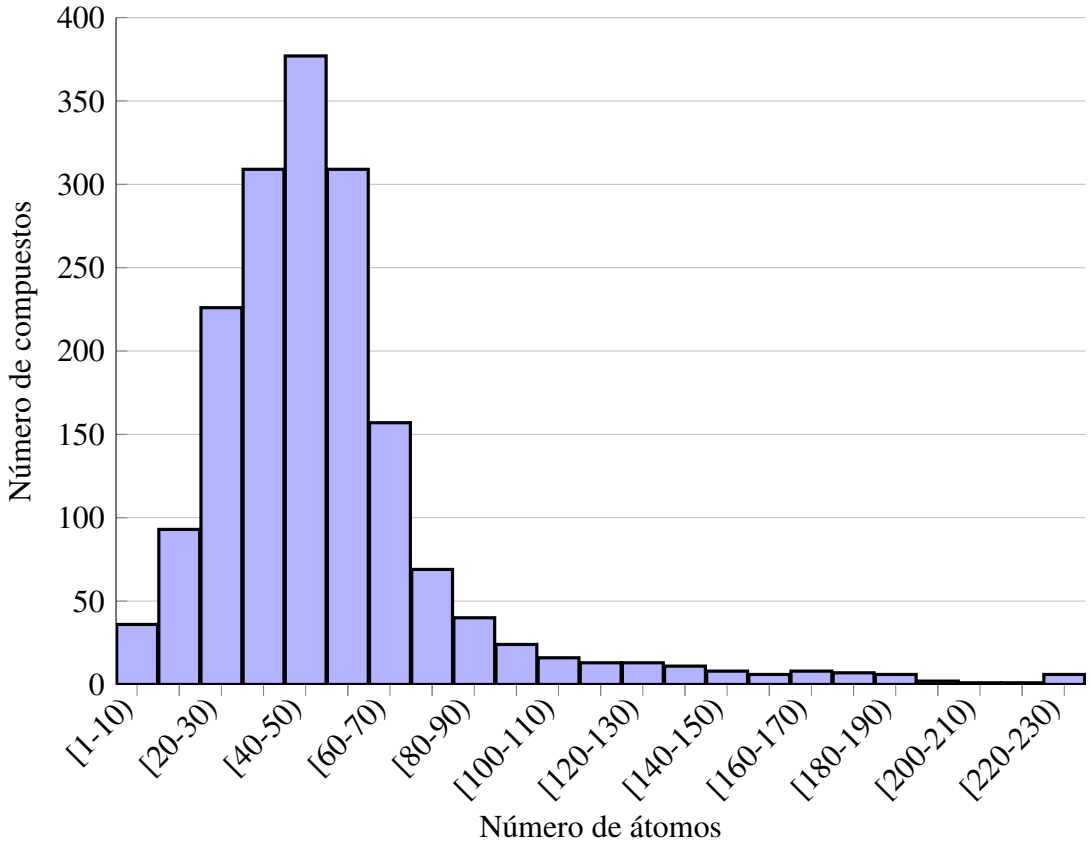


Figura 4.1: Número de compuestos incluidos en la base de datos de la FDA, agrupados por su número de átomos.

cada instancia seleccionando el mejor resultado obtenido para cada una de ellas. No obstante, es importante destacar que ejecutar varias veces una instancia en particular es solo una metodología para analizar la robustez del algoritmo; pero en el escenario del mundo real, OptiPharm solo necesita una sola ejecución para proporcionar resultados confiables. Por su parte, ROCS es determinista por lo que solo se ha ejecutado una vez.

La base de datos sobre la que se han comparado los métodos LBVS-Shape y LBVS-Electrostatic ha sido la FDA. Para este análisis, se realizó una selección de 50 compuestos de la siguiente manera: los compuestos en la base de datos se ordenaron por el número de átomos, incluyendo los hidrógenos, y luego se dividieron en 23 intervalos (ver figura 4.1). De cada uno de estos intervalos se eligió al menos un compuesto al azar siendo este número proporcional a la cantidad de compuestos en el intervalo.

En las siguientes secciones se presentan los experimentos realizados. En la sección 4.3 se ha realizado un estudio para saber cómo afecta el valor de H a los resultados finales desde el punto de vista de la similitud electrostática en la metodología LBVS-Shape. En la sección 4.4 se compara el rendimiento de ambas metodologías, LBVS-Shape y LBVS-Electrostatic. Como muestran los resultados, LBVS-Electrostatic parece obtener mejores soluciones que las obtenidas con el enfoque clásico de LBVS-Shape.

Tabla 4.1: Influencia del parámetro H en los resultados obtenidos por el método LBVS-Shape. Para cada valor de H, se muestran los siguientes valores medios de las 50 queries: posición en la clasificación de forma ($Av(Rk_R)$), número de átomos ($Av(nR)$), valor de similitud de forma ($Av(Tc_R)$), valor de evaluación de similitud electrostática ($Av(Tc_E^{Eval})$) y valor de similitud electrostática optimizado ($Av(Tc_E)$).

	Av				
H	Rk_R	nR	Tc_R	Tc_E^{Eval}	Tc_E
175	73	53	0.627	0.451	0.559
438	162	50	0.587	0.486	0.568
876	287	51	0.564	0.495	0.569
1313	324	50	0.559	0.497	0.570
1751	362	49	0.554	0.497	0.569

4.3 LBVS-Shape: Influencia del parámetro H en las predicciones.

La metodología LBVS-Shape basa sus predicciones en una preselección de los primeros H mejores compuestos en términos del valor de similitud de forma. Para demostrar su influencia, se ha aplicado el método LBVS-Shape sobre 50 queries cambiando el valor de H para que se seleccionen el 10 %, 25 %, 50 % y 100 % de los compuestos más parecidos durante la preselección. De este modo los valores de H se han fijado a 75, 438, 876, 1313 y 1751 compuestos.

La figura 4.2 ilustra un ejemplo de los pasos principales del método LBVS-Shape para la query DB01213 y $H = 1751$, es decir, el número total de compuestos en el conjunto de la base de datos FDA. Inicialmente, la query se compara con cada compuesto de la base de datos para obtener su posición óptima y el valor de similitud de forma correspondiente Tc_R. Como se mencionó anteriormente, en esta etapa se utiliza ROCS. Posteriormente, los compuestos se ordenan (Rk_R) en orden decreciente por Tc_R. Los H mejores compuestos se seleccionan y se evalúan para medir el valor de similitud electrostática correspondiente Tc_E^{Eval}. Se puede observar que la evaluación de la similitud electrostática considera la solución obtenida con la optimización de similitud de forma. El compuesto con el mayor Tc_E^{Eval}, llamado *BestComp*, se selecciona como la mejor predicción. Finalmente, como una etapa adicional y no considerada dentro del método LBVS-Shape, hemos calculado la superposición optimizada entre *BestComp* y query utilizando OptiPharm. En consecuencia, se proporciona también el valor correspondiente de Tc_E.

Para obtener una visión general de los resultados, se calcularon los valores medios de los *BestComp* encontrados para las 50 queries y cada valor de H, y se mostraron en la tabla 4.1. En particular, se calculó la posición media $Av(Rk_R)$ en la lista ordenada donde se ubicaron los *BestComp*, junto con lo siguiente: su número medio de átomos $Av(nR)$, su valor medio de similitud de forma $Av(Tc_R)$, su valor de similitud electrostática $Av(Tc_E^{Eval})$ cuando se evalúan y finalmente su similitud electrostática media cuando se optimizan $Av(Tc_E)$.

Como se puede ver, las predicciones parecen mejorar en términos de similitud electrostática a medida que aumenta el número H de moléculas seleccionadas en la lista ordenada (ver columnas $Av(Tc_E^{Eval})$ y $Av(Tc_E)$) . De acuerdo con estos resultados, la comparación posterior entre los métodos LBVS-Shape y LBVS-Electrostatic se ha llevado a cabo estableciendo $H = 1751$.

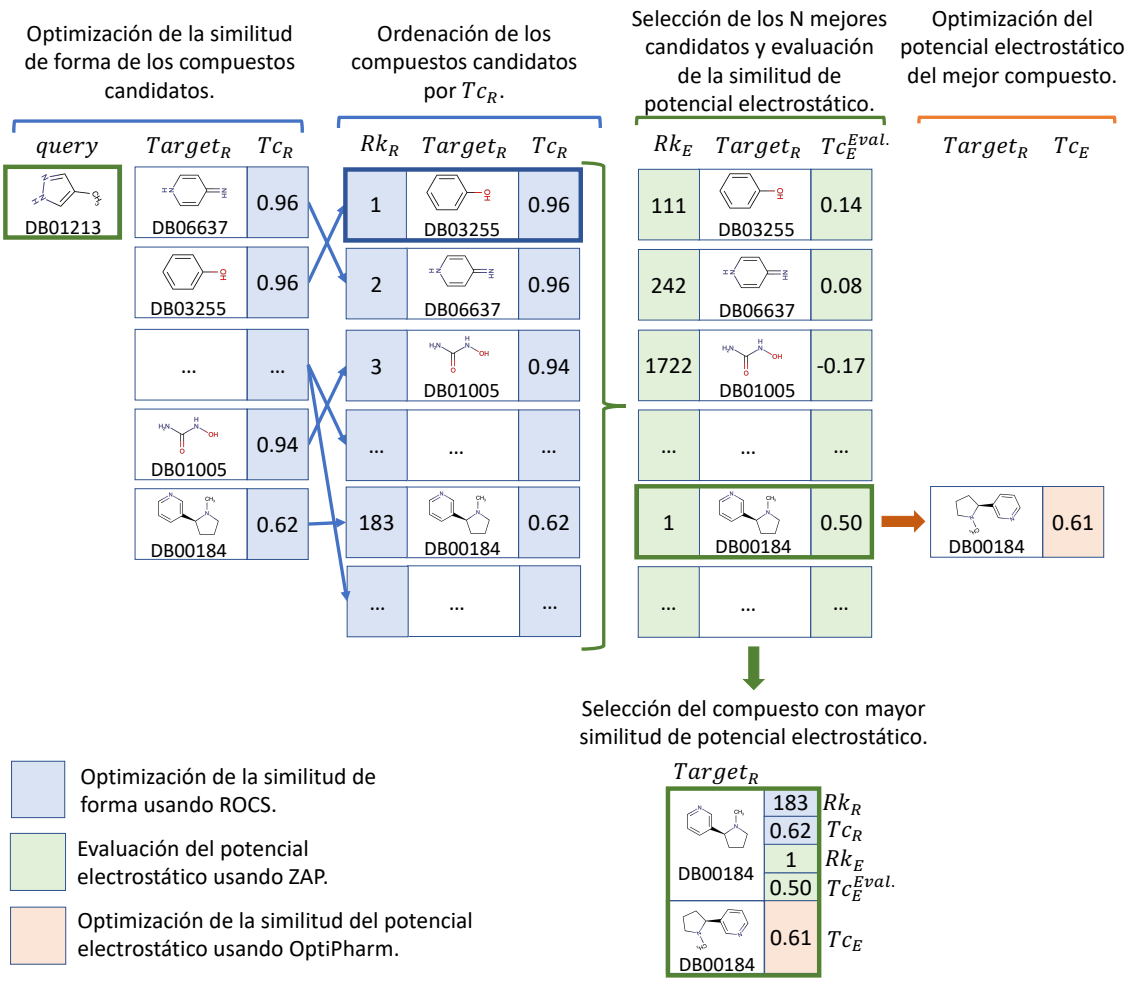


Figura 4.2: Un ejemplo del rendimiento del método LBVS-Shape para un caso particular donde query = DB01213 y $N = 1751$ usando la base de datos FDA.

4.4 LBVS-Shape versus LBVS-Electrostatic. Comparación de las predicciones obtenidas.

Como se explicó en la sección 4.1 en LBVS-Shape y LBVS-Electrostatic se siguen procedimientos distintos para obtener el compuesto más similar a otro de referencia. En esta sección se van a comparar ambas metodologías analizando los compuestos que obtiene cada una de ellas y sus correspondientes valores de similitud.

Previo a presentar los resultados, hay que tener en cuenta, al igual que sucede en el capítulo 3, que la comparación de un compuesto consigo mismo siempre alcanza el valor de similitud máximo, tanto para el potencial electrostático Tc_E como para la forma Tc_R. En consecuencia, estos resultados se eliminaron al clasificar los compuestos. En otras palabras, los compuestos dados como resultado no son los más similares, sino los segundos compuestos en la lista de clasificación. Además, como se mencionó anteriormente, el método LBVS-Shape se ha llevado a cabo considerando el número total de compuestos en la base de datos $H = 1751$ para aumentar la probabilidad de encontrar mejores predicciones.

Para entender la tabla resumen 4.3 donde se comparan los resultados de ambos métodos

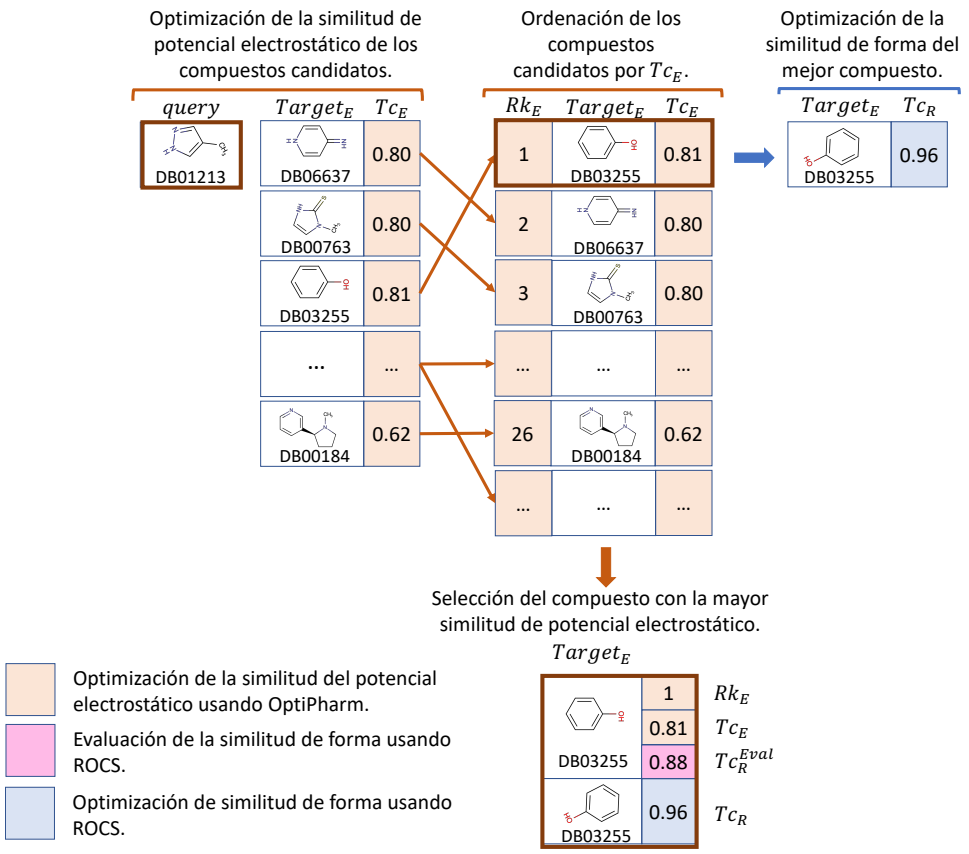


Figura 4.3: Un ejemplo del rendimiento del método electrostático LBVS-Electrostatic para un caso particular donde la query = DB01213 se compara con la base de datos de la FDA.

considerando 50 queries a continuación se estudia un caso particular. En concreto, se analiza la instancia query = DB01213. Este caso se utilizó también para ilustrar las etapas del método LBVS-Shape en la figura 4.2 y es el ejemplo que muestra el funcionamiento del método LBVS-Electrostatic (ver figura 4.3). Esta query ha sido seleccionada porque es pequeña y ayuda a ver las ideas principales de ambas metodologías muy fácilmente usando figuras. Las conclusiones inferidas de estos resultados pueden extrapolarse a cualquier otro compuesto query.

En la figura 4.3 primero se resuelve el problema de optimización para determinar la similitud electrostática, Tc_E, entre la query y cada target en la base de datos. Posteriormente, la lista de compuestos se ordena por el valor Tc_E y el que se encuentra en la primera posición, $Rk_E = 1$, se selecciona como la mejor predicción. Finalmente, para completar el estudio, se lleva a cabo la optimización para calcular la similitud de forma Tc_R entre el compuesto obtenido y la query.

Para una mayor claridad en la comparación, los resultados que se muestran en las figuras 4.2 y 4.3 se resumen en la tabla 4.2. El significado de las columnas, así como los valores en las tablas, son los explicados anteriormente y mostrados en cada figura. La última fila corresponde a los valores asociados con las mejores predicciones. Como se puede observar, cada método obtiene un compuesto diferente como mejor solución. LBVS-Shape obtiene como resultado a la molécula DB00184 con un $Tc_R = 0.621$ y un $Tc_E^{Eval} = 0.500$. Por su parte, LBVS-Electrostatic propone el compuesto DB03255 como el más similar a la query con $Tc_E = 0.810$ y $Tc_R^{Eval} = 0.880$. En consecuencia, el método LBVS-Electrostatic no solo ha obtenido un compuesto más similar en términos de potencial electrostático, sino también en forma. En la figura 4.4 se muestra la

Tabla 4.2: Resumen de los resultados obtenidos para los métodos LBVS-Shape y LBVS-Electrostatic para el compuesto query DB01213. La notación de columna, los colores y los resultados provienen de las figuras 4.2 y 4.3, es decir, mantienen el mismo significado que se mostró anteriormente para esas imágenes. La última fila indica los resultados asociados con la mejor solución seleccionada para cada método.

query		LBVS-Shape							LBVS-Electrostatic					
nombre	nA	Rk_R	$Target_R$	nR	Tc_R	Rk_E^{Eval}	Tc_E^{Eval}	Tc_E	Rk_E	$Target_E$	nE	Tc_E	Tc_R^{Eval}	Tc_R
DB01213	12	1	DB03255	13	0.963	111	0.140		1	DB03255	13	0.810	0.880	
		2	DB06637	13	0.961	242	0.081		2	DB06637	13	0.798	0.885	
		3	DB01005	9	0.943	1722	-0.172		3	DB00763	13	0.796	0.845	
					...							...		
		183	DB00184	26	0.621	1	0.500		110	DB00184	26	0.609	0.319	
					...							...		
DB01213	12	183	DB00184	26	0.621	1	0.500	0.609	1	DB03255	13	0.810	0.880	0.963

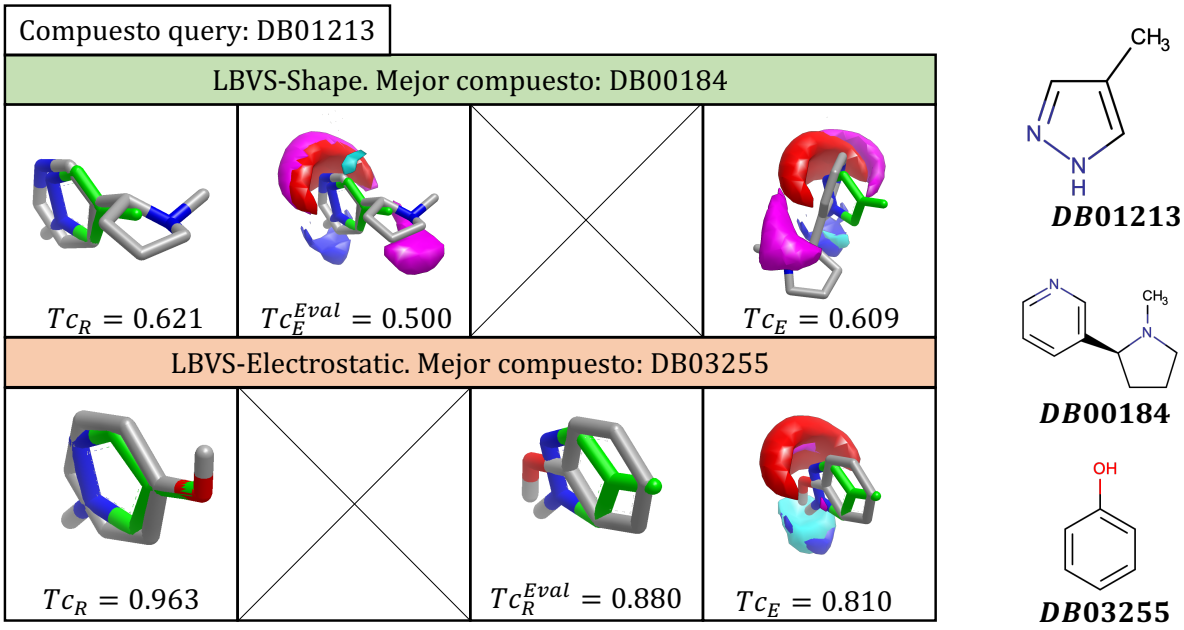


Figura 4.4: Resumen de resultados de LBVS-Shape y LBVS-Electrostatic para la query DB01213. El compuesto query es de color verde. Los campos electrostáticos del compuesto query son de color azul y rojo. Los mejores compuestos se muestran en gris y sus campos de potencial electrostático en cian y fucsia.

posición final para cada caso.

Una vez explicado el caso para la query DB01213, se muestra en la tabla 4.3 los resultados de las 50 queries seleccionadas previamente. Las columnas Rk_E^{Eval} y Rk_E se han eliminado en esta tabla porque sus valores son siempre 1. La última fila resume el valor medio de los resultados.

Como se puede observar, LBVS-Electrostatic obtiene un valor medio de $Tc_E = 0.738$, que es más alto que el dado por LBVS-Shape, $Tc_E^{Eval} = 0.497$. Se pueden obtener conclusiones similares al comparar los valores medios de Tc_E para ambos métodos. Además, cuando los resultados se analizan individualmente, podemos ver que LBVS-Electrostatic proporciona soluciones con valores Tc_E más altos que los logrados por LBVS-Shape. De hecho, en 48 de 50 casos, LBVS-Electrostatic obtiene un compuesto diferente al alcanzado por LBVS-Shape.

Tabla 4.3: Las filas se ordenan por el número de átomos de las queries. Para cada query, se sigue el mismo procedimiento explicado en la tabla 4.2. La última fila resume los valores medios de cada columna.

query		LBVS-Shape						LBVS-Electrostatic				
nombre	nA	Rk_R	Target_R	nR	Tc_R	Tc_E^{Eval}	Tc_E	Target_E	nE	Tc_E	Tc_R^{Eval}	Tc_R
DB00529	10	316	DB05266	35	0.496	0.437	0.593	DB00818	31	0.720	0.468	0.614
DB01213	12	182	DB00184	26	0.621	0.500	0.609	DB03255	13	0.810	0.880	0.963
DB00173	15	102	DB00851	23	0.792	0.546	0.536	DB01119	21	0.834	0.777	0.830
DB00172	17	24	DB00128	16	0.881	0.469	0.561	DB00677	25	0.699	0.690	0.769
DB00331	20	380	DB00961	40	0.598	0.599	0.697	DB01018	24	0.790	0.559	0.649
DB01119	21	513	DB00828	15	0.655	0.519	0.613	DB00173	15	0.832	0.779	0.829
DB02513	25	27	DB01275	20	0.872	0.526	0.569	DB06637	13	0.915	0.745	0.805
DB00915	28	125	DB00160	13	0.684	0.404	0.543	DB00478	34	0.946	0.673	0.924
DB01352	29	1	DB00306	32	0.926	0.947	0.983	DB00306	32	0.983	0.901	0.926
DB01365	30	180	DB01191	33	0.738	0.902	0.960	DB01626	26	0.964	0.628	0.824
DB00657	33	47	DB06770	16	0.788	0.396	0.517	DB01043	34	0.979	0.609	0.861
DB00478	34	30	DB00752	21	0.787	0.508	0.637	DB01043	34	0.957	0.615	0.879
DB01043	34	27	DB00945	21	0.765	0.400	0.478	DB00657	33	0.973	0.711	0.861
DB00380	35	601	DB00731	50	0.620	0.380	0.407	DB08971	56	0.505	0.435	0.655
DB00693	37	1034	DB04575	59	0.525	0.362	0.429	DB00692	40	0.454	0.391	0.783
DB09185	37	243	DB01233	43	0.722	0.839	0.506	DB09021	39	0.916	0.429	0.650
DB07615	40	71	DB04552	28	0.704	0.861	0.866	DB09218	28	0.892	0.610	0.574
DB09219	40	123	DB00321	44	0.698	0.347	0.329	DB00316	20	0.450	0.249	0.462
DB00674	42	279	DB00575	23	0.688	0.505	0.653	DB00514	45	0.662	0.415	0.695
DB00887	45	209	DB00232	31	0.642	0.401	0.454	DB01127	39	0.662	0.378	0.576
DB01198	45	273	DB00209	59	0.648	0.748	0.768	DB00123	25	0.894	0.334	0.491
DB01155	48	1	DB01165	46	0.858	0.671	0.818	DB01208	50	0.899	0.385	0.835
DB00246	50	467	DB00268	44	0.542	0.843	0.852	DB05271	48	0.877	0.391	0.604
DB00381	53	525	DB00573	32	0.577	0.285	0.278	DB00630	27	0.377	0.397	0.524
DB00876	54	576	DB01002	49	0.516	0.395	0.505	DB00774	28	0.532	0.276	0.524
DB09237	54	380	DB09092	44	0.580	0.759	0.824	DB08998	40	0.902	0.447	0.596
DB00254	55	1100	DB00271	28	0.521	0.626	0.836	DB00271	28	0.836	0.219	0.521
DB01268	57	902	DB09014	54	0.518	0.792	0.765	DB01409	48	0.883	0.421	0.564
DB01196	60	7	DB00783	44	0.741	0.397	0.385	DB08797	17	0.527	0.195	0.385
DB01621	66	274	DB00268	44	0.552	0.821	0.845	DB04861	55	0.867	0.330	0.454
DB09236	66	459	DB00607	51	0.509	0.406	0.438	DB00449	54	0.664	0.439	0.551
DB00632	69	537	DB00511	123	0.348	0.067	0.246	DB00898	9	0.997	0.126	0.137
DB08903	69	6	DB01433	58	0.621	0.840	0.867	DB01359	51	0.888	0.307	0.464
DB01419	70	380	DB09209	61	0.431	0.854	0.879	DB01611	51	0.933	0.291	0.423
DB00320	80	204	DB00438	59	0.515	0.367	0.396	DB00120	23	0.563	0.245	0.278
DB00728	91	1383	DB06204	40	0.399	0.688	0.761	DB09131	3	0.874	0.068	0.101
DB00503	98	655	DB00206	84	0.371	0.256	0.243	DB01144	22	0.401	0.180	0.280
DB01232	100	639	DB06480	52	0.389	0.691	0.741	DB09089	58	0.791	0.290	0.387
DB00309	110	385	DB01603	45	0.455	0.241	0.297	DB00319	63	0.467	0.267	0.534
DB04786	120	4	DB09158	82	0.377	0.424	0.708	DB09159	18	0.910	0.108	0.120
DB09114	130	117	DB00595	57	0.376	0.273	0.506	DB00583	26	0.876	0.183	0.190
DB06439	137	657	DB01628	39	0.383	0.336	0.425	DB00878	64	0.488	0.274	0.423
DB01078	140	34	DB00204	56	0.424	0.201	0.259	DB01085	31	0.540	0.169	0.211
DB01590	151	1037	DB01193	53	0.265	0.248	0.358	DB00653	6	0.529	0.070	0.100
DB04894	152	82	DB01199	87	0.361	0.348	0.484	DB09131	3	0.662	0.006	0.040
DB00403	167	325	DB04855	84	0.261	0.325	0.395	DB06335	49	0.575	0.120	0.198
DB00732	169	640	DB08967	52	0.222	0.236	0.353	DB00653	6	0.508	0.051	0.069
DB00050	194	7	DB01369	141	0.349	0.238	0.383	DB00516	19	0.385	0.059	0.080
DB06699	221	1465	DB01245	56	0.119	0.365	0.513	DB09131	3	0.642	0.013	0.029

Continúa en la siguiente página

query		LBVS-Shape						LBVS-Electrostatic				
nombre	nA	Rk_R	Target_R	nR	Tc_R	Tc_E^{Eval}	Tc_E	Target_E	nE	Tc_E	Tc_R^{Eval}	Tc_R
DB06219	229	69	DB01369	141	0.293	0.277	0.394	DB09131	3	0.670	0.009	0.021
media	74	362	-	49	0.554	0.497	0.569	-	31	0.738	0.372	0.505

Con respecto a la similitud de forma, es posible inferir que, en media, los métodos son equivalentes en términos de precisión de las predicciones, es decir, LBVS-Shape obtiene un valor medio de $Tc_R = 0.554$ mientras que LBVS-Electrostatic alcanza un valor medio de $Tc_R = 0.505$. Además, analizando los resultados obtenidos individualmente, podemos ver que en 2 de 50 casos, LBVS-Electrostatic ofrece predicciones mejores o equivalentes que las logradas por LBVS-Shape en términos de forma (ver columnas Tc_R en LBVS-Shape y Tc_R^{Eval} en LBVS-Electrostatic). Esto significa que existen casos en los que dos compuestos pueden ser muy similares en potencial electrostático, aunque pueden ser muy diferentes en forma. También implica que esas soluciones no se obtendrían nunca utilizando la metodología seguida por el método tradicional LBVS-Shape, ya que solo se enfocan en los compuestos con la mayor similitud en la forma.

Haciendo un estudio más detallado sobre los compuestos de menos de 50 átomos, es decir, los primeros 23 compuestos queries de la tabla 4.3, hay 5 casos en los que la diferencia es menor de 0.05 (DB00529, DB00173, DB00331, DB00915 y DB01352) y en otros 3 casos la diferencia es de 0.1 (DB01043, DB07615 y DB01268). Considerando los valores de estos 7 casos en los que la similitud de forma de LBVS-Electrostatic es más pequeña que la del LBVS-Shape con una diferencia media de 0.048, destaca la ganancia media en su similitud electrostática, que es de 0.271. En los compuestos grandes, donde se incluyen los 27 queries restantes, solo hay dos casos con características similares, que son los compuestos DB09236 con una diferencia de 0.07 y DB06699 con una diferencia de 0.013, ambos en similitud de forma. Viendo estos resultados, parece estar justificado el método LBVS-Electrostatic para dar compuestos candidatos a problemas con queries pequeñas.

Sin embargo, no todas las mejoras están relacionadas con los campos electrostáticos. La optimización del potencial electrostático utilizando OptiPharm también podría permitir encontrar una mejor solución en términos de forma. Los compuestos DB01119 y DB1213 en la tabla 4.3 son algunos ejemplos de ello. En el caso de la query DB01119, el mejor compuesto encontrado por LBVS-Shape es DB00828 con un $Tc_R = 0.655$ y un $Tc_E^{Eval} = 0.519$ mientras que el mejor compuesto de LBVS-Electrostatic es DB00173 que tiene un mejor Tc_E de 0.829 y de similitud de forma evaluada en la posición óptima alcanza al optimizar el potencial electrostático, $Tc_R^{Eval} = 0.779$.

4.5 Conclusiones

En este capítulo, se ha presentado un nuevo enfoque para resolver el problema LBVS basado en la similitud electrostática al que se le ha llamado LBVS-Electrostatic. Esta metodología se basa en la optimización directa de la similitud electrostática, el cual ha sido posible gracias a OptiPharm. Por el contrario, la metodología propuesta en la literatura, que se ha denominado LBVS-Shape, busca una sublista de los principales compuestos con la mayor similitud de forma

utilizando ROCS, para luego evaluar su similitud electrostática con ZAP. También se llevó a cabo un estudio para analizar la influencia del número de compuestos en dicha sublista. Como muestran los resultados, cuanto mayor sea el número de moléculas consideradas, mejor será la predicción obtenida en términos de similitud electrostática. A partir de esta conclusión, se realizó un estudio computacional para comparar el nuevo método LBVS-Electrostatic con el de la literatura LBVS-Shape. Para aumentar la probabilidad de encontrar buenas predicciones, LBVS-Shape se ha ejecutado teniendo en cuenta toda la base de datos antes de la evaluación de similitud electrostática. Aun así, LBVS-Electrostatic funciona mejor que LBVS-Shape, logrando mejores predicciones en el potencial electrostático para las 50 queries incluidas en el estudio. Con respecto a la similitud de forma, ambos métodos se comportan de manera similar, obteniendo en media compuestos con valores de similitud de forma similares. Es importante mencionar que la nueva metodología propuesta es novedosa, lo que significa que las predicciones obtenidas no han sido propuestas ni analizadas previamente.

Por otra parte, también se debe destacar el problema de desbordamiento encontrado en ZAP que implicó modificaciones en OptiPharm para su correcta evaluación. En el anexo B se explica con detalle el problema y la solución aplicada.

5. LBVS basado en la optimización multiobjetivo de la similitud de forma y el potencial electrostático

En este capítulo se realiza un estudio comparativo entre el enfoque monoobjetivo presentado en los capítulos anteriores y el multiobjetivo en el que se aplicará cribado virtual considerando la similitud de forma y el potencial electrostático conjuntamente. En la sección 5.1 se presenta el problema, así como las restricciones asociadas a él. En la sección 5.2 se realiza un análisis comparativo para seleccionar el algoritmo más apropiado para el enfoque multiobjetivo. En la sección 5.3 se realiza una comparativa entre las soluciones monoobjetivo y multiobjetivo utilizando para ello un caso de estudio. Finalmente en la sección 5.4 se recogen las conclusiones del capítulo.

5.1 Problema de optimización

Tal y como se mencionó en el capítulo 1, las técnicas de LBVS preseleccionan un reducido número de compuestos de entre los millones que puede contener una base de datos, para que las pruebas clínicas solo se realicen sobre estos. Los compuestos moleculares se caracterizan por un conjunto de propiedades. En consecuencia, cuanta más información se considere en las comparaciones, mejores podrán ser las predicciones propuestas. Con esta hipótesis, en este capítulo se propone abordar el cribado virtual de compuestos desde un enfoque multiobjetivo. Así, a diferencia de los capítulos 3 y 4 donde el cribado se realiza en base a una única propiedad (forma o potencial electrostático), a continuación se define un problema en que el ambas funciones objetivo se consideran simultanéamente.

Desde un punto de vista matemático, el modelo se define en el problema 5.1. $\boldsymbol{P}_{Mqt}$ puede entenderse como un problema de maximización en el que dado un compuesto query q y una base de datos de compuestos DB, encontrar para cada compuesto $t \in DB$, todo el conjunto eficiente o conjunto óptimo de Pareto (ver definición 1.2.4), es decir, todos los puntos que son eficientes mediante la optimización de las variables de decisión (Θ, c_1, c_2 y Δ) dentro de sus

límites que fueron definidos en el capítulo 2. Es importante recordar que estos límites se calculan dinámicamente para cada pareja de compuestos y dependen de los tamaños de estos últimos.

Problema 5.1

$$\boldsymbol{P}_{Mqt} = \begin{cases} \text{máx} & Tc_{Sqt}(\Theta_t, c1_t, c2_t, \Delta_t) \\ \text{máx} & Tc_{Eqt}(\Theta_t, c1_t, c2_t, \Delta_t) \\ \text{s.t.} & \Theta_t \in [0, 2\pi](\text{rad}), \\ & c1_t, c2_t \in [(\underline{-box_{t_x}, -box_{t_y}, -box_{t_z}}), (\overline{box_{t_x}, box_{t_y}, box_{t_z}})](\text{Å}), \\ & \Delta_t \in [(\underline{-\Delta_{qt_x}, -\Delta_{qt_y}, -\Delta_{qt_z}}), (\overline{\Delta_{qt_x}, \Delta_{qt_y}, \Delta_{qt_z}})](\text{Å}) \end{cases} \tag{5.1}$$

5.2 Algoritmo de optimización multiobjetivo

Para resolver el problema 5.1, se puede utilizar una gran variedad de algoritmos multiobjetivo generalistas existentes en la literatura. En esta tesis se ha optado por utilizar la plataforma jMetal [140]. jMetal es un framework que incluye un conjunto de metaheurísticas multiobjetivo. Algunas de ellas son GDE3 [141], Epsilon-IBEA [142], MOCell [143], MOEAD [144], NSGAII [145], OMOPSO [146], SMPSO [147] and SPEA2 [148]. Gracias a su estructura modular, la implementación de un problema puede ser integrada en cada uno de los algoritmos y ejecutada para así comprobar sus soluciones de una forma rápida y precisa. En esta tesis se han utilizado las versiones 1.5.1 de Python y 1.7 de Java.

En primera instancia se ha realizado un estudio computacional que ha permitido seleccionar aquel algoritmo que se adapta mejor a nuestro problema. Este estudio ha consistido en la optimización de 81 targets frente a 19 queries. Para seleccionar los compuestos, primero se ha dividido en grupos los 823 compuestos rígidos de la base de datos DrugBank según el número de átomos de las moléculas. Dicha división se muestra en la figura 5.1. De cada intervalo se ha seleccionado un compuesto al azar obteniendo de esta forma las 19 queries. Para elegir los 81 targets, se han agrupado los compuestos con el mismo número de átomos y se ha seleccionado uno de cada grupo aleatoriamente. Tanto los compuestos queries como targets tienen un número de átomos comprendido en el intervalo $[2, 106]$.

En relación a los algoritmos, estos han sido ejecutados con la configuración propuesta por los autores en sus respectivos trabajos, a excepción del número de evaluaciones y el tamaño de la población de la lista de soluciones. Para analizar la convergencia de los algoritmos, se ha probado con tres valores distintos en el parámetro que define el número de evaluaciones. En concreto, 15000, 25000 y 35000. El tamaño de la población se ha fijado a 300 puntos para todos los casos. Con este análisis se busca encontrar el algoritmo multiobjetivo más adecuado para abordar el problema 5.1 seleccionando aquel que tenga el mejor compromiso entre calidad de la solución, evaluada mediante el hipervolumen (ver sección 1.2.3.1), y el tiempo, medido en segundos.

En la tabla 5.1 se resumen los resultados obtenidos. En concreto se muestra el valor medio del hipervolumen Hv y el tiempo de ejecución en segundos T de todas las queries frente a los targets para cada algoritmo. En cada columna se ha resaltado el mejor resultado obtenido. Como puede observarse, independientemente del algoritmo, tanto el hipervolumen como el tiempo de cómputo crece según lo hace el número de evaluaciones. En términos de hipervolumen, o

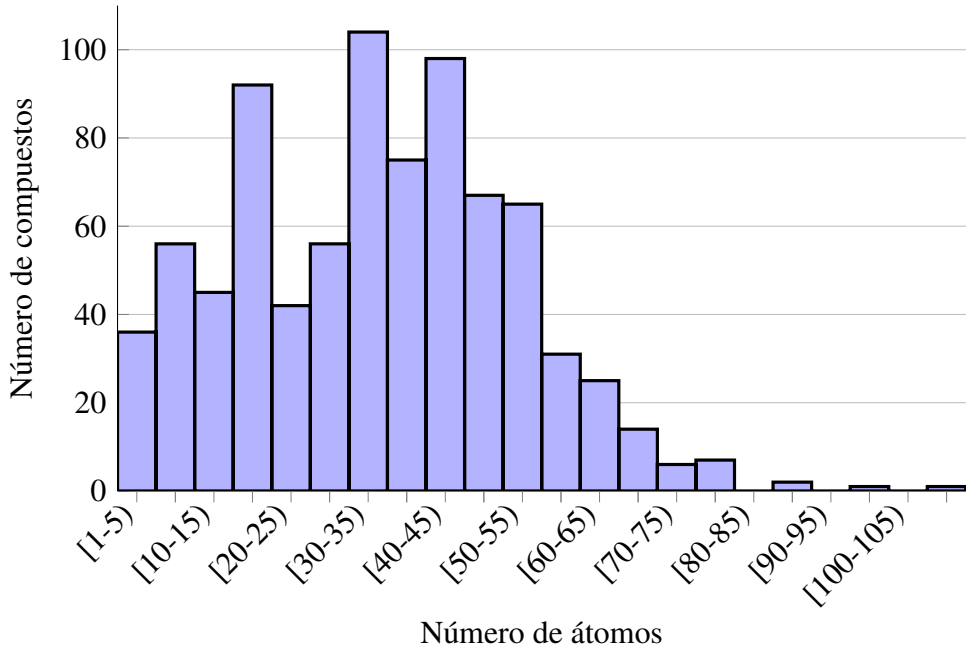


Figura 5.1: Número de compuestos rígidos incluidos en la base de datos DrugBank, agrupados por su número de átomos.

lo que es lo mismo, calidad de la solución, el algoritmo MOEA/D parece ser el que mejores resultados ofrece. No obstante, en cuanto a tiempo de ejecución, el algoritmo SMPSO parece ser el más rápido. En vista a los resultados obtenidos y buscando un equilibrio entre la calidad de la soluciones y el tiempo computacional, nos hemos centrado en los resultados obtenidos con la configuración de 25000 evaluaciones. El algoritmo seleccionado para analizar la metodología multiobjetivo ha sido MOEA/D, por ser el que mejor valor de hipervolumen ha obtenido. Por tanto, este será el algoritmo que consideraremos en las comparaciones posteriores.

5.3 Comparación entre las predicciones monoobjetivo y multiobjetivo: caso de estudio

En esta sección, para ilustrar las diferencias en las predicciones obtenidas desde un enfoque monoobjetivo y multiobjetivo, se ha seleccionado un caso de estudio. El algoritmo monoobjetivo elegido ha sido OptiPharm cuyas predicciones se compararan a las de MOEA/D que se ha seleccionado en base a los resultados del análisis previo. No obstante, se ha realizado un estudio computacional más amplio de otros 7 compuestos queries adicionales evaluados frente a la base de datos FDA y del que se han inferido conclusiones similares.

Los algoritmos se han configurado para que exploren en profundidad el espacio de búsqueda y obtener, de este modo, los mejores resultados posibles. Así, se ha utilizado la configuración robusta para OptiPharm que tan buenos resultados ha obtenido en los capítulos 3 y 4; y para MOEA/D se ha mantenido la configuración establecida por sus autores salvo el número de evaluaciones que se ha fijado a 200000, las mismas que OptiPharm, y el número de soluciones que mantiene durante el proceso de optimización, que es de 300. Para ambos algoritmos se han considerado los átomos de hidrógeno.

Tabla 5.1: Valor medio de hipervolumen $Av(Hv)$ y tiempo de ejecución (en segundos) $Av(T)$ para cada algoritmo y cada número de evaluaciones de los resultados obtenidos al optimizar 81 targets respecto de 19 queries. El mejor resultado de cada columna está resaltado.

	Evaluaciones					
	15000		25000		35000	
Algoritmo	$Av(Hv)$	$Av(T)$	$Av(Hv)$	$Av(T)$	$Av(Hv)$	$Av(T)$
Epsilon-IBEA	0.476	281	0.479	504	0.482	769
GDE3	0.446	187	0.458	318	0.464	487
MOCell	0.454	181	0.456	300	0.458	469
MOEA/D	0.475	182	**0.483**	301	**0.484**	473
NSGAII	0.473	187	0.478	301	0.480	486
OMOPSO	0.469	167	0.476	241	0.476	421
SMPSO	**0.476**	**164**	0.480	**227**	0.482	**420**
SPEA2	0.474	333	0.479	581	0.480	1125

Si disponemos de un algoritmo monoobjetivo y queremos obtener soluciones con alta similaridad en ambos descriptores de manera simultánea, el procedimiento a seguir sería el siguiente. En primer lugar se ejecuta Optipharm con la función Tc_S para el cálculo de la similtud en forma, se ordenan las soluciones y se evalúan con la función objetivo del potencial electrostático obteniendo Tc_E^{Eval}. Del mismo modo, para obtener los compuestos más similares en potencial electrostático, podríamos utilizar OptiPharm para optimizar la similitud de potencial electrostático y tras ordenar los compuestos por dicha función objetivo, evaluarlos con Tc_S para obtener así su valor de similitud de forma Tc_S^{Eval}. Con ambos listados calculados, se deben seleccionar los compuestos más apropiados, para lo cual se hace un filtro por umbral. Dado que no existe una herramienta que asegure encontrar el mejor compuesto y por tanto, que pueda utilizarse para definir un umbral universal, el umbral de referencia considerado en los siguientes experimentos estará delimitado por el valor del mejor compuesto encontrado. En concreto, se preseleccionarán todos aquellos compuestos con un valor de similitud superior al 80 % del valor de similitud del mejor compuesto encontrado.

Tras aplicar el filtro en los listados obtenidos por OptiPharm, se han reducido los 1751 compuestos a 228 para la similitud de forma y 8 para la del potencial electrostático cuyos resultados se pueden ver en las tablas 5.2 y 5.3, respectivamente aunque por conveniencia, para la tabla 5.2 no se presentan todos los resultados, sino una muestra reducida de ellos. El número de compuestos para el primer listado sigue siendo alto y observando los resultados de la tabla 5.2 se puede ver que sus valores de similitud electrostática, Tc_E^{Eval}, son mayoritariamente muy bajos y por tanto carecen de interés. Esto sucede en las 217 de las 228 propuestas, donde el valor de Tc_E^{Eval} es inferior a 0.25. En consecuencia, se ha realizado otro filtrado seleccionando solo aquellos compuestos cuyo valor sea superior a 0.50 en ambos descriptores. Estos están resaltados en las propias tablas 5.2 y 5.3. Como se puede observar, los compuestos que se seleccionan finalmente mediante los algoritmos monoobjetivo son el DB01165 y el DB00218.

Conseguidos los resultados de los algoritmos monoobjetivo, a continuación obtendremos las mejores predicciones según el algoritmo multiobjetivo. La metodología para elegirlos utilizando un algoritmo multiobjetivo difiere de la de los monoobjetivo. En concreto, esta se ilustra en la figura 5.2. El proceso comienza obteniendo para cada pareja de compuestos query y target su frente de Pareto correspondiente, es decir, se resuelve para cada dupla el problema 5.1 con

Tabla 5.2: Listado de compuestos encontrados para el compuesto query DB01155 por OptiPharm optimizando la similitud de forma (Tc_S) y cuyo valor es superior al 80 % del mejor compuesto encontrado. Esto es $0.773x0.80 = 0.618$. Dado que se obtienen 228 compuestos en ese intervalo, se han mostrado los 17 primeros compuestos y los dos últimos. Para cada compuesto se ha calculado también su valor de similitud de potencial electrostático, Tc_E^{Eval}.

query	Rk_S	Target	Tc_S	Tc_E^{Eval}
DB01155	1	DB01208	0.773	-0.112
	2	DB01113	0.771	0.045
	3	DB01059	0.767	0.130
	4	**DB01165**	**0.766**	**0.693**
	5	DB00467	0.763	0.061
	6	DB00618	0.752	0.153
	7	DB01137	0.735	0.055
	8	DB00537	0.725	0.058
	9	DB00487	0.717	0.135
	10	DB09047	0.716	-0.144
	11	DB09137	0.711	-0.128
	12	DB01426	0.711	0.265
	13	**DB00218**	**0.708**	**0.690**
	14	DB09093	0.702	0.065
	15	DB00931	0.700	0.107
	16	DB00240	0.699	0.081
	17	DB01190	0.698	0.220
	...	...	...	...
	227	DB00713	0.619	0.074
	228	DB01039	0.619	-0.018
media	-	-	0.719	0.128

Tabla 5.3: Listado de compuestos encontrados para el compuesto query DB01155 por OptiPharm optimizando el potencial electrostático (Tc_E) y cuyo valor es superior al 80 % del mejor compuesto encontrado. Esto es $0.930x0.80 = 0.744$. Para cada compuesto se ha calculado también su valor de similitud de forma, Tc_S^{Eval}.

query	Rk_E	Target	Tc_E	Tc_S^{Eval}
DB01155	**1**	**DB01165**	**0.930**	**0.548**
	2	DB06771	0.926	0.436
	3	**DB00218**	**0.912**	**0.626**
	4	DB01208	0.910	0.432
	5	DB09047	0.883	0.264
	6	DB00729	0.860	0.409
	7	DB00462	0.842	0.423
	8	DB00278	0.797	0.350
media	-	-	0.883	0.436

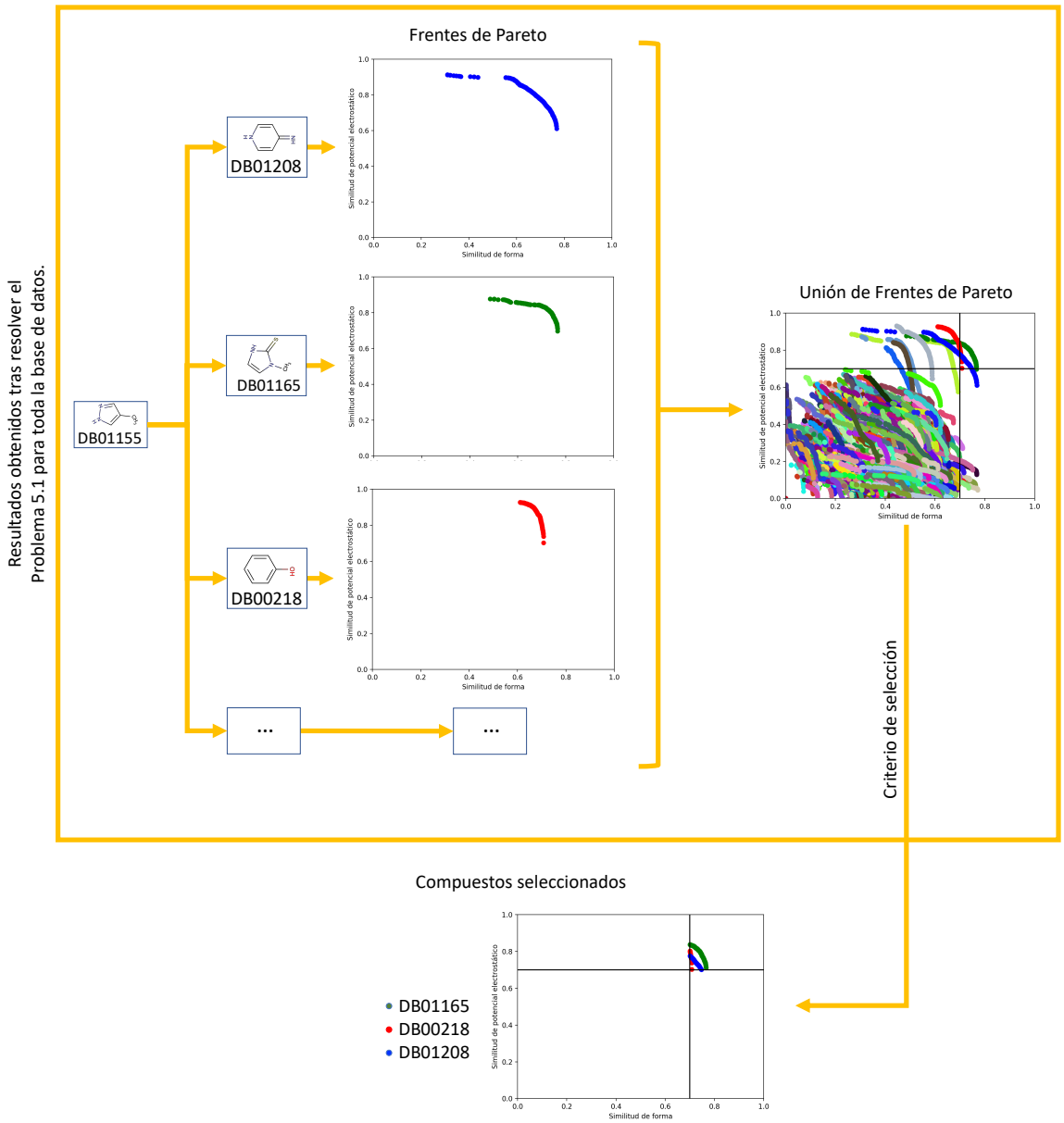


Figura 5.2: Un ejemplo del rendimiento del método multiobjetivo para un caso particular.

MOEA/D. Una vez obtenidos todos los frentes, se agrupan en un único archivo. Sobre este se aplican los criterios de selección deseados por el decisor, reduciendo de este modo el número de compuestos finales. Para comparar en igualdad de condiciones, los umbrales que se han aplicado a estos resultados son los mismos que los utilizados para las soluciones monoobjetivo. Tras aplicar el criterio de selección, solo 3 compuestos han sido seleccionados pues tenían soluciones del frente de Pareto con valores superiores a los umbrales establecidos, tanto de similitud de forma como de potencial electrostático. En la tabla 5.4 se resumen los compuestos obtenidos por MOEA/D a lo que se han añadido las soluciones obtenidas por OptiPharm con las dos funciones objetivo para facilitar la comparación.

En vista de los resultados, queda reflejada la ventaja de una metodología multiobjetivo frente a la monoobjetivo. En este caso estudiado, se ha encontrado un tercer compuesto, el DB01208, que no habría sido posible predecir desde el punto de vista monoobjetivo pues si bien, este está incluido en el listado de compuestos ofrecido por las soluciones monoobjetivo tras el primer

Tabla 5.4: Resultados obtenidos por los algoritmos monoobjetivo y multiobjetivo para el compuesto query DB01155 después de aplicar el filtrado definido por los umbrales.

query	Algoritmo	Target	Tc_S	Tc_S^{Eval}	Tc_E	Tc_E^{Eval}
DB01155	Monoobjetivo Forma	DB01165	0.766	-	-	0.693
		DB00218	0.708	-	-	0.690
	Monoobjetivo Electrostático	DB01165	-	0.548	0.930	-
		DB00218	-	0.626	0.912	-
	Multiobjetivo	DB01208	0.747	-	0.776	-
		DB01165	0.766	-	0.837	-
		DB00218	0.708	-	0.803	-

filtro, su bajo valor en el descriptor evaluado no permite diferenciar este compuesto de otros que tienen también valores similares. Además, las soluciones monoobjetivo solo ofrecen una única solución para cada pareja query-target mientras que los multiobjetivo dependen de su configuración, que en este experimento ha consistido en permitir ofrecer hasta 300 soluciones distintas. Esto es muy interesante pues dependiendo de los intereses finales, se puede seleccionar una solución u otra sin tener que volver a ejecutar ningún experimento.

5.4 Conclusiones

En este capítulo se han comparado las predicciones obtenidas por un enfoque monoobjetivo con el multiobjetivo. Para el primero se ha utilizando el algoritmo OptiPharm que tan buenos resultados ha dado en los problemas presentados en los capítulos 3 y 4. Por su parte, el algoritmo multiobjetivo ha sido seleccionado a partir de un estudio comparativo entre los diferentes algoritmos presentes en jMetal. A vista de los resultados, se ha elegido MOEA/D pues es el algoritmo que mejor compromiso ha obtenido entre la calidad de las soluciones, medidas mediante el hipervolumen, y el tiempo de ejecución, en segundos. Respecto a la comparativa entre OptiPharm y MOEA/D, se ha utilizado la base de datos FDA configurando ambos algoritmos con configuraciones equiparables y que permitieran explorar el espacio de búsqueda exhaustivamente. Además, para obtener una mejor precisión, se consideraron los hidrógenos para ambas funciones objetivo. Los resultados han mostrado que la metodología multiobjetivo presenta algunas ventajas respecto a las monoobjetivo en cuanto se optimizan las dos funciones objetivo simultáneamente. Traducido a nuestro problema, la propuesta multiobjetivo es capaz de encontrar compuestos que de otra forma serían descartados por los monoobjetivo.

III

Conclusiones y Trabajo Futuro

6. Conclusiones y trabajo futuro

6.1 Español

Esta tesis se enmarca dentro del conjunto de técnicas computacionales que permiten acelerar el proceso de descubrimiento y desarrollo de fármacos. En ese sentido se han propuesto nuevas herramientas y metodologías para hacer frente a uno de los principales problemas como es la preselección de candidatos que se utilizarán posteriormente en las pruebas clínicas. Los algoritmos diseñados han obtenido mejores resultados que los algoritmos de la literatura y las nuevas metodologías proponen soluciones de una mayor calidad y en un menor tiempo de ejecución. Todos estos avances han permitido en nuestros experimentos y permitirán en un futuro ofrecer compuestos que de otra forma nunca se hubieran propuesto.

La primera contribución de esta tesis ha sido el diseño de un nuevo algoritmo que hemos llamado OptiPharm. OptiPharm es un algoritmo evolutivo memético diseñado específicamente para problemas de cribado virtual. En concreto, se puede adaptar a cualquier problema en el que la función objetivo esté estrechamente relacionada con la posición 3D de los compuestos a evaluar. Además, es un algoritmo de optimización global y parametrizable. Lo primero implica que puede explorar todo el espacio de búsqueda a diferencia de los algoritmos de la literatura basados mayormente en optimizaciones locales. Esto, en un principio, podría tener el efecto negativo al necesitar, a priori, más tiempo para obtener los resultados, sin embargo, al ser parametrizable se puede configurar adaptando su ejecución a los diferentes problemas. El resultado que se ha obtenido de esta combinación es conseguir un algoritmo capaz de explorar más espacio de búsqueda en menor tiempo que los algoritmos de la literatura y con resultados de calidad similar o superior.

Optipharm se ha aplicado a dos problemas. El cribado virtual basado en similitud de forma y el del potencial electrostático. Respecto al problema de la similitud de forma, dado que no existe una solución conocida para estos problemas, en primer lugar se comprobó la calidad de

OptiPharm utilizando la base de datos Maybridge. Esta base de datos es útil porque contiene una gran cantidad de compuestos de tamaño similares entre sí. Esto facilita encontrar compuestos con alto grado de similitud de forma a otro compuesto de referencia. Los resultados demostraron que OptiPharm es capaz de encontrar buenas soluciones para los casos en los que efectivamente existían compuestos de tamaño similar, pero no solo eso, sino que en aquellos donde existían un reducido número de compuestos, también ha encontrado soluciones de buena calidad.

Posteriormente se comparó OptiPharm con WEGA, el actual algoritmo de referencia de la literatura. El análisis se han realizado utilizando las bases de datos FDA, DUD y DUD-E. Aprovechando la parametrización de OptiPharm, este se ejecutó con dos configuraciones distintas: una en la que se explora más exhaustivamente el espacio de búsqueda y otra en la que se prima la velocidad para obtener los resultados. Por su parte, WEGA no es configurable por lo que se ejecutó con su única configuración. Además, como este último no considera los hidrógenos, OptiPharm se configuró de forma similar para obtener los mismos valores de la función objetivo. Los resultados fueron favorables a nuestro algoritmo pues con las dos configuraciones se podía elegir entre obtener resultados similares a los de WEGA pero 5 veces más rápido o explorar en profundidad el espacio de búsqueda y encontrar mejores resultados.

Por último, se realizó un estudio donde se analizó la influencia de considerar o no los átomos de hidrógenos. Si bien WEGA y el resto de las herramientas en la literatura no consideran dichos átomos en las evaluaciones, pensamos que desde el punto de vista químico sí es interesante. Los resultados refuerzan nuestra hipótesis, consiguiendo en primer lugar resultados distintos al considerar o no los átomos de hidrógeno y en segundo lugar, el modelo con hidrógenos obtenía compuestos más similares. En consecuencia, concluimos que eliminar los hidrógenos se entiende como una pérdida de información.

El segundo problema abordado en esta tesis ha sido el cribado virtual basado en ligados que optimiza la similitud del potencial electrostático. Aquí se propuso una nueva metodología que se comparó con el método tradicional de la literatura. Este último es ampliamente utilizado y consiste en obtener el listado de los compuestos más similares en forma y posteriormente, de un subconjunto reducido de ellos, obtener el compuesto más similar en potencial electrostático sin realizar ninguna optimización sobre el valor de esta segunda característica. Por su parte, el método que nosotros proponemos consiste en la optimización directa de la similitud de potencial electrostático. Desde el punto de vista de la optimización, resulta complicado seleccionar el mejor compuesto en base a una característica cuando los esfuerzos para conseguir el compuesto más similar se destinan a otra propiedad. Como muestran los resultados, nuestro método encuentra mejores soluciones y propone compuestos que desde la metodología tradicional serían descartados desde el inicio. Pero no todo está relacionado con el potencial electrostático sino que con nuestro método se han encontrado también compuestos con mejor valor de similitud de forma que aquellos encontrados por la metodología tradicional. Aquí se muestran claramente las ventajas, una vez más, de OptiPharm respecto de los algoritmos de optimización local de la literatura.

En este problema, también se analizó la influencia de la cantidad de compuestos que son optimizados en forma y posteriormente evaluados en potencial electrostático. Se demostró que el procedimiento de seleccionar un pequeño porcentaje, como se viene haciendo en la literatura, es contraproducente y en consecuencia, no debería realizarse ningún filtrado sino considerar todos los compuestos.

Durante los experimentos además se detectó un problema que existía en la evaluación de la similitud del potencial electrostático. En concreto con el programa ZAP, en el que dadas ciertas posiciones espaciales de los compuestos, su evaluación de la función objetivo es incorrecta.

El último problema que se ha abordado en esta tesis ha sido desde el punto de vista multiobjetivo. En concreto se ha realizado una comparación entre las predicciones de un algoritmo monoobjetivo y otro multiobjetivo. Para la selección de este último, se realizó un estudio comparativo entre distintos algoritmos de la literatura que convencionalmente obtienen buenos resultados. Con los resultados obtenidos, se consideró que MOEA/D es el más adecuado para este problema. Además, se abordó el prefiltrado de compuestos desde un punto de vista práctico. En ese sentido, se cribaron aquellos compuestos no interesantes del listado final. En relación a los resultados finales, ha quedado demostrado la necesidad del uso de algoritmos multiobjetivo, que incorporen o evalúen más información de forma simultanea pues permiten encontrar compuestos que con técnicas monoobjetivo serían descartados o su análisis en ensayos clínicos implicaría un aumento considerable de costes.

Todos los experimentos de esta tesis han sido posible llevarlos a cabo gracias al diseño de técnicas de balanceo de carga que han aprovechado de la mejor forma las ventajas de la HPC de los centros de supercomputación utilizados. Si bien su uso no ha estado exento de problemas y su funcionamiento fue inicialmente difícil, la ventaja en reducción de tiempo de cómputo ha sido muy importante. Además, relacionado con esto y de cara a permitir a la comunidad utilizar libremente OptiPharm sin realizar ninguna configuración, este se ha integrado en BRUSELAS (`http://bio-hpc.eu/software/Bruselas/`) [149], un servidor web en el que se pueden realizar baterías de experimentos utilizando varias aplicaciones software, y su código fuente está disponible bajo las condiciones del anexo C. Por último, en relación a las técnicas de paralelismo, estas se han aplicado en trabajos relacionados.

Finalmente y gracias a las publicaciones asociadas a los avances realizados, la comunidad internacional se ha hecho eco de las ventajas de OptiPharm. En la figura 6.1 se muestran representados en verde los países de cuya procedencia son las solicitudes que se han recibido hasta la fecha de escritura de esta tesis para utilizar OptiPharm. Pertenecen tanto a empresas como instituciones académicas siendo un total de 12 solicitudes de distintos países: Alemania, Canadá, China, Estados Unidos, España, Francia, Polonia, Egipto, India y Singapur. Además, cabe destacar que OptiPharm se está usando actualmente para un proyecto relacionado con el COVID-19 con intereses comerciales internacionales y en otro proyecto con una empresa europea con la que se han encontrado efectos que podrían ser relevantes para el tratamiento de ciertas condiciones médicas y que se encuentra todavía en fase de experimentación y cuyos detalles no se pueden mencionar debido a contratos de confidencialidad establecidos con las diferentes empresas. En este segundo proyecto, además de los compuestos encontrados, se sigue trabajando para proponer nuevas moléculas candidatas.

El trabajo futuro que queda planteado en esta tesis se divide en varios frentes. En primer lugar, hay que considerar la incorporación de nuevos descriptores a las comparaciones. De esta forma los compuestos serán seleccionados con un mayor conocimiento. Esta selección puede hacerse tanto desde el punto de vista monoobjetivo como multiobjetivo. El segundo punto a tener en cuenta es la flexibilidad de los compuestos. En todo momento los compuestos tienen una estructura fija, sin embargo y como se mencionó en el capítulo 1, los átomos dentro de las moléculas puede desplazarse en ciertos límites. Un estudio sobre ello y su incorporación en las

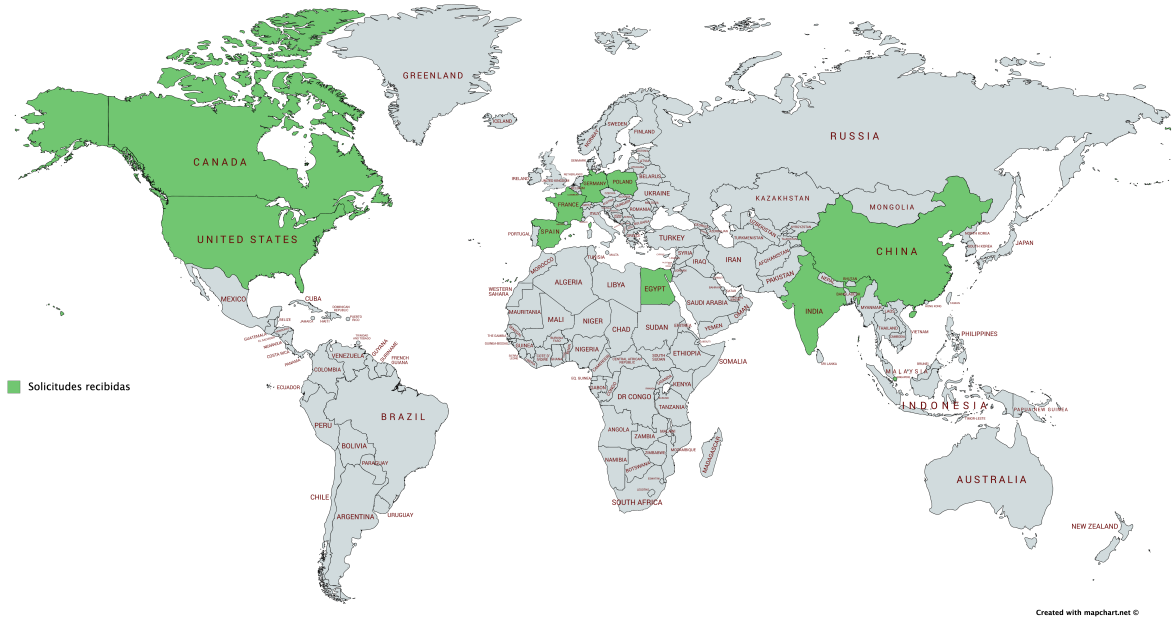


Figura 6.1: Mapamundi donde están resaltados en verde los países de los que se ha recibido solicitudes sobre OptiPharm.

funciones objetivo descritas añadirían un valor adicional a las comparaciones pues estas serían más precisas. Finalmente, se ha visto la cantidad de tiempo necesario por los algoritmos para obtener los resultados, no por la evaluación individual de dos compuestos sino por la cantidad de estos últimos. En consecuencia, queda planteada también una línea en la que los cálculos se realicen sobre GPUs.

6.2 English

This thesis is part of the set of computational techniques that allow to accelerate the process of drug development and discovery. In this regard, new tools and methodologies have been proposed to face one of the main problems, which is the pre-selection of candidates to be used later in clinical trials. The algorithms designed have obtained better results than the algorithms in the literature and the new methodologies propose solutions of higher quality and in a shorter runtime. All these advances have allowed us to offer compounds in our experiments that otherwise would never have been proposedand will also allow us to do so in the future.

The first contribution of this thesis has been the design of a new algorithm that we have called OptiPharm. OptiPharm is a memetic evolutionary algorithm designed specifically for virtual screening problems. In particular, it can be adapted to any problem where the objective function is closely related to the 3D position of the compounds to be evaluated. Moreover, it is a global and parameterizable optimization algorithm. The fact that it is global means it can explore the whole search space unlike the algorithms in the literature based mostly on local optimizations. This, at first, could have the negative effect of needing, a priori, more time to obtain the results. However, being parameterizable means it can be configured by adapting its execution to different problems. The result obtained from this combination is an algorithm capable of exploring more

search space in less time than the algorithms in the literature and with results of similar or higher quality.

Optipharm has been applied to two problems. Virtual screening based on shape and electrostatic potential similarities. Regarding the problem of shape similarity, since there is no known solution to these problems, first the quality of OptiPharm was checked using the Maybridge database. This database is useful because it contains a large number of compounds of similar size to each other. This makes it easy to find compounds with a high degree of shape similarity to another reference compound. The results showed that OptiPharm is able to find good solutions for cases where there were indeed compounds of similar size, but not only that, in those where there were a small number of compounds, it also found good quality solutions.

Subsequently, OptiPharm was compared with WEGA, the current reference algorithm in the literature. The analysis was performed using the FDA, DUD and DUD-E databases. Taking advantage of the parameterization of OptiPharm, it was executed with two different configurations: one in which the search space is explored more exhaustively and another in which the speed to obtain the results is prioritized. On the other hand, WEGA is not configurable so it was executed with its only configuration. In addition, as it does not consider the hydrogens, OptiPharm was configured in a similar way to obtain the same values of the objective function. The results were favorable to our algorithm because, with the two configurations, you could choose either obtaining similar results to WEGA but being 5 times faster or explore in depth the search space finding better results.

Finally, a study was carried out where the influence of whether or not to consider the hydrogen atoms was analyzed. Although WEGA and the rest of the tools in the literature do not consider these atoms in the evaluations, we think that it is of interest from a chemical point of view . The results proved us to be right, firstly obtaining different results when considering the hydrogen atoms to when they were not; and secondly, the model with hydrogen atoms obtained more similar compounds. Consequently, we conclude that eliminating hydrogen can be understood as a loss of information.

The second problem addressed in this thesis is the ligand based virtual screening that optimizes the similarity of the electrostatic potential. Here, a new methodology was proposed and compared with the traditional method in the literature. The latter is widely used and consists of obtaining the list of the most similar compounds in shape and subsequently, from a reduced subset of them, to obtain the most similar compound in electrostatic potential without optimizating the value of this second characteristic. On the other hand, the method that we propose consists of the direct optimization of electrostatic potential similarity. From the point of view of optimization, it is complicated to select the best compound based on an objective function when the efforts to obtain the most similar compound are directed to another function. As the results show, our method finds better solutions and proposes compounds that from traditional methodology would be discarded from the very beginning. Not everything is related to electrostatic potential but with our method we have also found compounds with a better similarity value than those found by traditional methodology. Here the advantages of OptiPharm over the local optimization algorithms in the literature are clearly shown once again. As part of this problem, the influence of the amount of compounds that are optimized in shape and subsequently evaluated in electrostatic potential was also analyzed. It was demonstrated that the procedure of selecting a small percentage, as is being done in the literature, is counterproductive and consequently, no filtering should be done,

as such, considering all the compounds.

During the experiments, an existing problem in the evaluation of the similarity of the electrostatic potential was also detected. In particular, with the software ZAP. Its evaluation of the objective function is incorrect in certain positions of the compounds.

The last problem addressed in this thesis was from the multiobjective point of view. In particular, a comparison between the predictions of a monoobjective and a multiobjective algorithm was made. For the selection of the latter, a comparative study was performed between different algorithms in the literature that conventionally obtain good results. With the results obtained, it was considered that MOEA/D is the most suitable for this problem. In addition, the pre-filtering of compounds was approached from a practical point of view. As such, those compounds from the final list that were not of interest, were screened. In relation to the final results, the need for the use of multiobjective algorithms has been demonstrated. They incorporate or evaluate more information simultaneously because they allow compounds to be found that with monoobjective techniques would be discarded or their analysis in clinical trials would imply a considerable increase in costs.

All the experiments in this thesis were able to be carried out thanks to the design of load balancing techniques that have taken advantage of the benefits of the HPC from the supercomputing centers used. Although their use was not problem-free and their operation was initially difficult, the advantage in reducing computing time was very important. In addition, on a related note and in order to allow the community to freely use OptiPharm without configuration, it was integrated into BRUSELAS (`http://bio-hpc.eu/software/Bruselas/`) [149], a web server where you can perform batteries of experiments using various software applications, and its source code is available under the conditions detailed in annex C. Finally, concerning parallelism techniques, these have been applied in related works.

Finally and thanks to the publications associated with the progress made, the international community has echoed the benefits of OptiPharm. In figure 6.1 represented in green are the countries of origin for the requests that have been received to date to use OptiPharm. They belong both to companies and academic institutions with a total of 12 applications from different countries: Canada, China, United States, Germany, Spain, France, Poland, Egypt, India and Singapore. In addition, it should be noted that OptiPharm is currently being used for a project related to COVID-19 with international commercial interests as well as in another project with a European company in which effects have been found that could be relevant to the treatment of certain medical conditions and which is still in the experimental phase. In this second project, in addition to the compounds found, work continues to propose new candidate molecules.

The future work that is set out in this thesis is divided into several fronts. Firstly, the incorporation of new descriptors to the comparisons must be considered. In this way the compounds will be selected with a greater knowledge. This selection can be made both from a monoobjective and a multiobjective point of view. The second point to take into account is the flexibility of the compounds. At all times the compounds have a fixed structure, although as mentioned in chapter 1, the atoms within the molecules can move in certain limits. A study of this and its incorporation into the objective functions described would add value to the comparisons as they would become more accurate. Finally, we have seen the amount of time needed by the algorithms to obtain the results, not because of the individual evaluation of two compounds, but because of the quantity

of the latter. Consequently, a line has also been proposed in which the calculations are made on GPUs.

IV Anexos

A. Cálculo de la curva ROC

En este anexo se explica brevemente el cálculo del Área bajo la curva ROC (Característica Operativa del Receptor) (AUC, *Area Under the Curve ROC (Receiver Operating Characteristic)*). Para un mayor detalle, se recomienda ver el trabajo de Fawcett [150].

A.1 Definición

Una curva ROC es una técnica de visualización, organización y clasificación basada en los valores obtenidos de un experimento. Su primera aplicación fue en el campo de la teoría de señales para diferenciar aquellas que eran reales de las que no [151, 152]. Posteriormente se ha aplicado a otras disciplinas como el ámbito médico para la toma de decisiones en las pruebas de diagnóstico [153]. En la inteligencia artificial, no fue hasta el trabajo de Spackman [154] cuando se empezó a usar las curvas ROC para el aprendizaje automático y la evaluación y comparación de algoritmos. Desde ese momento, se ha extendido hasta llegar a ser una práctica común en la comparación de algoritmos y métodos en diversas áreas.

Antes de explicar como se calcula, es necesario conocer algunos conceptos:

- Activos. Los compuestos activos son compuestos con una alta actividad hacia el compuesto de referencia. El umbral exacto de actividad sobre el cual un compuesto se considera activo es arbitrario, pero los compuestos a menudo se consideran activos si tienen valores de actividad IC50 [155].
- Inactivos. Los compuestos inactivos son compuestos que se conocen que tienen una baja actividad (o ninguna actividad) hacia el compuesto de referencia por lo que deben evitarse.
- Señuelos (Decoys). Los compuestos señuelos son compuestos que se parecen a los compuestos activos pero que aceptamos como inactivos. Los señuelos generalmente se obtienen buscando compuestos que tengan descriptores físicos similares a los compuestos activos,

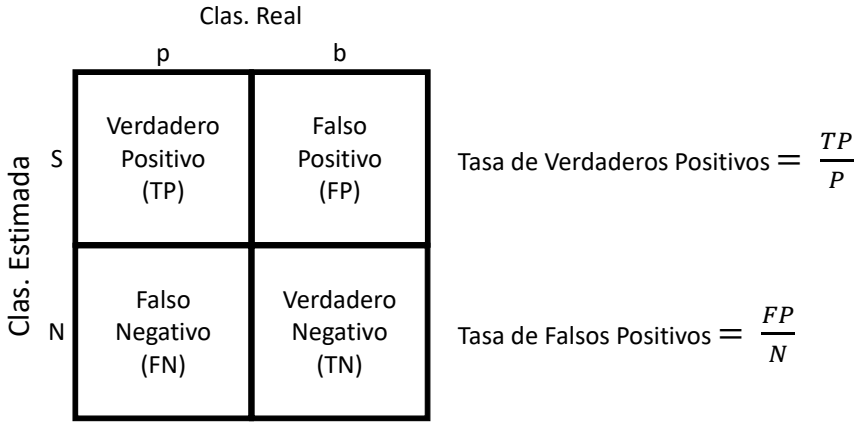


Figura A.1: Matriz de confusión.

pero que son químicamente diferentes.

A.2 Cálculo del Área bajo la Curva ROC (ROC AUC)

La base de la curva ROC parte de un conjunto de elementos en el que hay activos y señuelos y estos son conocidos. Sin embargo, para el algoritmo a evaluar, todos los compuestos son iguales y debe ser este quien clasifique los compuestos como activos y señuelos. Según la clasificación que realice el algoritmo y teniendo en cuenta la clase verdadera de los elementos, se genera una tabla de contingencia o matriz de confusión (ver figura A.1). Este concepto apareció por primera vez en [156].

Como se puede observar en la figura, dependiendo de la clasificación de los elementos, puede dar lugar a 4 categorías distintas:

- *Verdadero positivo*: elemento que es verdadero y se ha clasificado como verdadero.
- *Verdadero negativo*: elemento falso que se ha clasificado como falso.
- *Falso positivo*: elemento falso clasificado como verdadero.
- *Falso negativo*: elemento verdadero clasificado como falso.

Conociendo el número total de elementos que han sido evaluados, las cuatro categorías son complementarias dos a dos, por lo que solamente dos de ellas son necesarias para realizar los análisis de clasificación. Las categorías seleccionadas para crear la curva ROC son los verdaderos positivos y los falsos positivos (ver figura A.2). Las dos dimensiones del gráfico en el que la curva se representa toman valores en el intervalo $[0,1]$. En este cuadro se representa el ratio de verdaderos positivos y falsos positivos que ha clasificado el algoritmo.

Además, es una práctica común añadir una diagonal desde el punto $(0,0)$ al $(1,1)$. Los puntos que se situen en el triángulo superior, representando en verde, como es el punto P_1 indica que el algoritmo ha clasificado correctamente ese elemento. Si el punto, por el contrario, está situado en el triángulo rojo inferior, como es el caso de P_3 significa que la clasificación es incorrecta. Por último, si el punto se sitúa sobre la diagonal (P_2), la conclusión que se obtiene es que el algoritmo clasifica aleatoriamente los elementos como verdaderos o falsos.

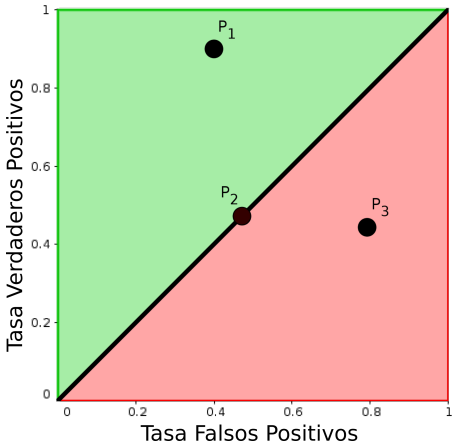
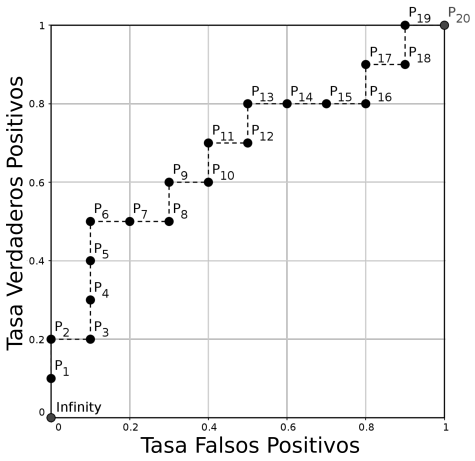


Figura A.2: Un gráfico ROC mostrando 3 valores discretos.

ID	Clase	Valor	ID	Clase	Valor
P_1	p	.9	P_{11}	p	.4
P_2	p	.8	P_{12}	n	.39
P_3	n	.7	P_{13}	p	.38
P_4	p	.6	P_{14}	n	.37
P_5	p	.55	P_{15}	n	.36
P_6	p	.54	P_{16}	n	.35
P_7	n	.53	P_{17}	p	.34
P_8	n	.52	P_{18}	n	.33
P_9	p	.51	P_{19}	p	.30
P_{10}	n	.505	P_{20}	n	.1

(a) Elementos de entrada.



(b) Curva ROC generada.

Figura A.3: Un ejemplo de generación de una curva ROC.

En el problema de cribado virtual, un punto no es suficiente para obtener la precisión de clasificación de un algoritmo sino que se obtiene un conjunto de ellos. En concreto, dado el conjunto de elementos y la clasificación de cada uno de ellos obtenida por el algoritmo, la curva ROC se construye iterando por todos los elementos secuencialmente y obteniendo cuantos verdaderos positivos se encuentran en las primeras posiciones. Este mecanismo se describe en el algoritmo 3 y un ejemplo de su ejecución se puede ver en la figura A.3 original de [150]. En la subfigura A.3a se muestran tres columnas donde se describe el *ID* de los elementos, la *Clase* a la que pertenencen y el *Valor* obtenido por un algoritmo de clasificación. Los 20 elementos se dividen en 10 positivos y 10 negativos. En la subfigura A.3b se muestra la curva ROC generada por la ejecución del algoritmo utilizando los elementos de entrada. De forma resumida, el procedimiento se explica a continuación. Se itera desde el primer elemento y si el compuesto es positivo, se incrementa la tasa de verdaderos positivos, en caso contrario, se hace lo propio en la tasa de falsos positivos. Para obtener una clasificación perfecta, los 10 elementos positivos deberían haber ocupado las 10 primeras posiciones en la tabla.

Una vez generada la curva con todos los puntos, se calcula el área bajo ella. Cuanto más próximo sea el valor a 1, mejor habrá sido la clasificación. Si el valor es cercano a 0, significa que la clasificación es completamente errónea. Por último, si el valor es cercano a 0.5, la conclusión

Algoritmo 3 Método eficiente de generación de puntos ROC

Input: L, el conjunto de muestras; $f(i)$, la estimación del clasificador probabilístico de que la muestra i sea positiva; P y N, número de muestras positivas y negativas.

Output: R, una lista de puntos ROC que incrementan por tasa de falsos positivos.

Required: $P > 0$ y $N > 0$.

1: $L_{sorted} = L$ ordenados por valores de f decrecientes
2: $FP = TP = 0$
3: $R = ()$
4: $f_{prev} = -$ínf
5: $i = 1$
6: **while** $i \leq |L_{sorted}|$ **do**
7: **if** $f(i) \neq f_{prev}$ **then**
8: insertar($\frac{FP}{N}, \frac{TP}{P}$) en R
9: $f_{prev} = f(i)$
10: **if** $L_{sorted}[i]$ es una muestra positiva **then**
11: $TP = TP + 1$
12: **else** ▷ Es una muestra negativa
13: $FP = FP + 1$
14: $i = i + 1$
15: insertar($\frac{FP}{N}, \frac{TP}{P}$) en R ▷ Esto es (1,1)

que se obtiene es que el algoritmo clasifica azarosamente los elementos. El cálculo de este área se realiza mediante el algoritmo 4.

Algoritmo 4 Cálculo del área bajo la curva ROC

Input: L, el conjunto de muestras; $f(i)$, la estimación del clasificador probabilístico de que la muestra i sea positiva; P y N, número de muestras positivas y negativas.

Output: A, el área bajo la curva ROC.

Required: $P > 0$ y $N > 0$.

1: $L_{sorted} = L$ ordenados por valores de f decrecientes
2: $FP = TP = 0$
3: $A = 0$
4: $f_{prev} = -$ínf
5: $i = 1$
6: **while** $i \leq |L_{sorted}|$ **do**
7: **if** $f(i) \neq f_{prev}$ **then**
8: $A = A + TRAPEZOID\_AREA(FP, FP_{prev}, TP, TP_{prev})$
9: $f_{prev} = f(i)$
10: $FP_{prev} = FP$
11: $TP_{prev} = TP$
12: **if** i es una muestra positiva **then**
13: $TP = TP + 1$
14: **else** ▷ Es una muestra negativa
15: $FP = FP + 1$
16: $i = i + 1$
17: $A = A + TRAPEZOID\_AREA(FP, FP_{prev}, TP, TP_{prev})$
18: $A = A/(P \times N)$ ▷ Se escala a la unidad cuadrada
19: **procedure** TRAPEZOID_AREA($X1$, $X2$, $X3$, $X4$)
20: $Base = |X1 - X2|$
21: $Height_{avg} = (Y1 + Y2)/2$
22: **return** $Base \times Height_{avg}$

B. Problemas de precisión de ZAP

En este anexo se reporta un error encontrado durante la experimentación realizada por OptiPharm para el cribado virtual basado en la similitud del potencial electrostático. ZAP es un kit de herramientas ofrecido por OpenEye y es ampliamente utilizado en la literatura para calcular dicha similitud [121-133, 157, 158].

En la búsqueda de nuevas soluciones candidatas, OptiPharm puede separar progresivamente dos compuestos de entrada intentando escapar de los óptimos locales y explorar en profundidad el espacio de búsqueda. De hecho, es posible analizar casos donde no existe solapamiento físico entre las moléculas de entrada.

Durante el análisis de los resultados, descubrimos que existen situaciones en las que ZAP puede desbordarse. En la figura B.1 se muestra un ejemplo de ello. Aquí, el compuesto query DB01365[1] permanece fijo en la parte izquierda mientras que el compuesto DB00459[2] está en el lado derecho en tres posiciones espaciales diferentes, diferenciándose cada uno por el color. El valor de similitud de potencial electrostático entre el compuesto query y el target representado en rojo es de 1. Esto indicaría que ambos compuestos tienen un potencial electrostático idéntico. Si este compuesto se desplaza 0.5Å hacia la izquierda, pasando a estar en la posición del compuesto de color cian, el valor de similitud de potencial electrostático se reduce a 0.38. Pero es más, si en vez de desplazar el compuesto hacia la izquierda, se desplaza 0.5Å hacia la derecha, ocupando la posición del compuesto rosa, el valor de similitud es de 0. En consecuencia, se puede deducir que la posición del compuesto target cian es el lugar exacto en el que las funciones de ZAP se desbordan produciendo un resultado erróneo.

Detectado este problema en los cálculos de OptiPharm, se ha resuelto considerando soluciones inviables todas aquellas cuya similitud de forma sea 0, es decir, no existe solapamiento entre

[1] https://go.drugbank.com/drugs/DB01365
[2] https://go.drugbank.com/drugs/DB00459

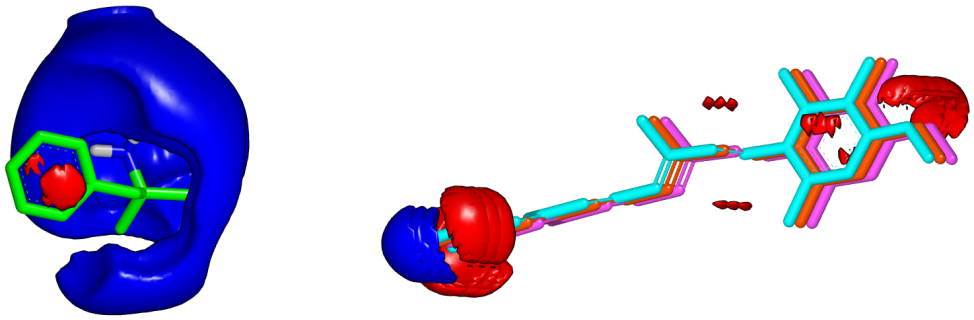


Figura B.1: El compuesto DB01365 está pintado en verde. El compuesto DB00459 está representado en con tres diferentes colores: cian, rojo y rosa. Los campos electrostáticos se imprimen en azul oscuro y rojo con el software VIDA [159].

ambos compuestos. En consecuencia, estas soluciones no se considerarán durante el proceso de optimización.

C. Disponibilidad del software y datos

En este anexo se declara la disponibilidad del software y de las bases de datos utilizadas en esta tesis.

C.1 OptiPharm

El ejecutable de OptiPharm se puede usar en el servidor BRUSELAS [160] que está accesible desde la web `http://bio-hpc.eu/software/bruselas/`. El código fuente se encuentra bajo las siguientes condiciones en un repositorio GitLab.

- Nombre del proyecto: OptiPharm_ES.
- Página de inicio del proyecto: `https://hpca.ual.es/optipharm/ES/`
- Repositorio de código fuente del proyecto: `https://gitlab.hpca.ual.es/savins/optipharm_es`
- Sistema operativo (s): Linux y MacOS.
- Lenguaje de programación: C ++.
- Licencia: Mozilla Public License 2.0.
- Cualquier restricción de uso por parte de no académicos: se necesita licencia, contacto con los autores.

C.2 Bases de datos y software de terceros

Las bases de datos y software utilizado para los diferentes estudios pertenecen a sus autores y el acceso a ellas depende de las restricciones aplicables.

Índice de figuras

Índice de tablas

Bibliografía

[1] P. Mehta, D. F. McAuley, M. Brown, E. Sánchez, R. S. Tattersall y J. J. Manson, "COVID-19: consider cytokine storm syndromes and immunosuppression," *The Lancet*, volumen 395, número 10229, páginas 1033-1034, 2020.

[2] S. Baize, D. Pannetier, L. Oestereich, T. Rieger, L. Koivogui, N. F. Magassouba, B. Soropogui, M. S. Sow, S. Këita, H. De Clerck, A. Tiffany, G. Dominguez, M. Loua, A. Traoré, M. Kolié, E. R. Malano, E. Heleze, A. Bocquin, S. Mély, H. Raoul, V. Caro, D. Cadar, M. Gabriel, M. Pahlmann, D. Tappe, J. Schmidt-Chanasit, B. Impouma, A. K. Diallo, P. Formenty, M. Van Herp y S. Günther, "Emergence of Zaire Ebola Virus Disease in Guinea," *New England Journal of Medicine*, volumen 371, número 15, páginas 1418-1425, 2014.

[3] L. R. Petersen, D. J. Jamieson, A. M. Powers y M. A. Honein, "Zika Virus," *New England Journal of Medicine*, volumen 374, número 16, páginas 1552-1563, 2016.

[4] T. Liu, D. Lu, H. Zhang, M. Zheng, H. Yang, Y. Xu, C. Luo, W. Zhu, K. Yu y H. Jiang, "Applying high-performance computing in drug discovery and molecular simulation," *National Science Review*, volumen 3, número 1, páginas 49-63, 2016.

[5] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik y A. Zhavoronkov, "Reinforced Adversarial Neural Computer for de Novo Molecular Design," *Journal of Chemical Information and Modeling*, volumen 58, número 6, páginas 1194-1204, 2018.

[6] P. Ertl, "Cheminformatics Analysis of Organic Substituents: Identification of the Most Common Substituents, Calculation of Substituent Properties, and Automatic Identification of Drug-like Bioisosteric Groups," *Journal of Chemical Information and Computer Sciences*, volumen 43, número 2, páginas 374-380, 2003.

[7] A. M. Virshup, J. Contreras-Garcıa, P. Wipf, W. Yang y D. N. Beratan, "Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possi-

ble Drug-Like Compounds," *Journal of the American Chemical Society*, volumen 135, número 19, páginas 7296-7303, 2013.

[8] D. Horvath, "A Virtual Screening Approach Applied to the Search for Trypanothione Reductase Inhibitors," *Journal of Medicinal Chemistry*, volumen 40, número 15, páginas 2412-2423, 1997.

[9] T. Scior, A. Bender, G. Tresadern, J. L. Medina-Franco, K. Martınez-Mayorga, T. Langer, K. Cuanalo-Contreras y D. K. Agrafiotis, "Recognizing pitfalls in virtual screening: A critical review," *Journal of Chemical Information and Modeling*, volumen 52, número 4, páginas 867-881, 2012.

[10] A. Kumar y K. Y. Zhang, "Hierarchical virtual screening approaches in small molecule drug discovery," *Methods*, volumen 71, páginas 26-37, 2015.

[11] I.-L. Lu, C.-F. Huang, Y.-H. Peng, Y.-T. Lin, H.-P. Hsieh, C.-T. Chen, T.-W. Lien, H.-J. Lee, N. Mahindroo, E. Prakash, A. Yueh, H.-Y. Chen, C. M. V. Goparaju, X. Chen, C.-C. Liao, Y.-S. Chao, J. T.-A. Hsu y S.-Y. Wu, "Structure-Based Drug Design of a Novel Family of PPAR γ Partial Agonists: Virtual Screening, X-ray Crystallography, and in Vitro/in Vivo Biological Activities," *Journal of Medicinal Chemistry*, volumen 49, número 9, páginas 2703-2712, 2006.

[12] J. L. Stark y R. Powers, "Application of NMR and Molecular Docking in Structure-Based Drug Discovery," en, 2011, páginas 1-34.

[13] G. M. Morris y M. Lim-Wilby, "Molecular Docking," en, 2008, páginas 365-382.

[14] N. Brooijmans e I. D. Kuntz, "Molecular Recognition and Docking Algorithms," *Annual Review of Biophysics and Biomolecular Structure*, volumen 32, número 1, páginas 335-373, 2003.

[15] N. S. Pagadala, K. Syed y J. Tuszynski, "Software for molecular docking: a review," *Biophysical Reviews*, volumen 9, número 2, páginas 91-102, 2017.

[16] M. Karelson, *Molecular descriptors in QSAR/QSPR*. Wiley-Interscience New York, 2000, volumen 230.

[17] P. C. D. Hawkins, A. G. Skillman y A. Nicholls, "Comparison of Shape-Matching and Docking as Virtual Screening Tools," *Journal of Medicinal Chemistry*, volumen 50, número 1, páginas 74-82, 2007.

[18] E. Lionta, G. Spyrou, D. K. Vassilatis y Z. Cournia, "Structure-based virtual screening for drug discovery: principles, applications and recent advances.," *Current Topics in Medicinal Chemistry*, volumen 14, número 16, 2014.

[19] P. D. Lyne, "Structure-based virtual screening: an overview," *Drug discovery today*, volumen 7, número 20, páginas 1047-1055, 2002.

[20] M. Wójcikowski, P. J. Ballester y P. Siedlecki, "Performance of machine-learning scoring functions in structure-based virtual screening," *Scientific Reports*, volumen 7, número 1, página 46 710, 2017.

[21] M. A. Johnson y G. M. Maggiora, *Concepts and applications of molecular similarity*. Wiley, 1990.

[22] R. Todeschini y V. Consonni, *Handbook of molecular descriptors*. John Wiley & Sons, 2008, volumen 11.

[23] A. Pozzan, "Molecular Descriptors and Methods for Ligand Based Virtual High Throughput Screening in Drug Discovery," *Current Pharmaceutical Design*, volumen 12, número 17, páginas 2099-2110, 2006.

[24] A. Kumar y K. Y. J. Zhang, "Advances in the Development of Shape Similarity Methods and Their Application in Drug Discovery," *Frontiers in Chemistry*, volumen 6, página 315, 2018.

[25] J. A. Grant, M. A. Gallardo y B. T. Pickup, "A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape," *Journal of Computational Chemistry*, volumen 17, número 14, páginas 1653-1666, 1996.

[26] B. B. Masek, A. Merchant y J. B. Matthew, "Molecular shape comparison of angiotensin II receptor antagonists," *Journal of Medicinal Chemistry*, volumen 36, número 9, páginas 1230-1238, 1993.

[27] M. L. Connolly, "Computation of molecular volume," *Journal of the American Chemical Society*, volumen 107, número 5, páginas 1118-1124, 1985.

[28] J. A. Grant y B. T. Pickup, "A Gaussian Description of Molecular Shape," *The Journal of Physical Chemistry*, volumen 99, número 11, páginas 3503-3510, 1995.

[29] X. Yan, J. Li, Z. Liu, M. Zheng, H. Ge y J. Xu, "Enhancing Molecular Shape Comparison by Weighted Gaussian Functions," *Journal of Chemical Information and Modeling*, volumen 53, número 8, páginas 1967-1978, 2013.

[30] M. J. Vainio, J. S. Puranen y M. S. Johnson, "ShaEP: Molecular Overlay Based on Shape and Electrostatic Potential," *Journal of Chemical Information and Modeling*, volumen 49, número 2, páginas 492-502, 2009.

[31] C. Böttcher, O. V. Belle y B. Belle, *Theory of electric polarization*. Elsevier Scientific Pub. Co, 1974.

[32] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles*, volumen 37, páginas 241-272, 1901.

[33] T. Sterling y J. J. Irwin, "ZINC 15 –Ligand Discovery for Everyone," *Journal of Chemical Information and Modeling*, volumen 55, número 11, páginas 2324-2337, 2015.

[34] H. E. Pence y A Williams, "ChemSpider: An Online Chemical Information Resource," *Journal of Chemical Education*, volumen 87, número 11, páginas 1123-1124, 2010.

[35] V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, A. Tang, G. Gabriel, C. Ly, S. Adamjee, Z. T. Dame, B. Han, Y. Zhou y D. S. Wishart, "DrugBank 4.0: shedding new light on drug metabolism," *Nucleic Acids Research*, volumen 42, número D1, página D1091, 2014.

[36] N. M. O' Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch y G. R. Hutchison, "Open Babel: An open chemical toolbox," *Journal of Cheminformatics*, volumen 3, número 1, página 33, 2011.

[37] T. A. Halgren, "Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94," *Journal of Computational Chemistry*, volumen 17, número 5-6, páginas 490-519, 1996.

[38] *Tripos Mol2 File Format*.

[39] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox y M. Wilson, "DrugBank 5.0: a major update to the DrugBank database for 2018," *Nucleic Acids Research*, volumen 46, número D1, páginas D1074-D1082, 2018.

[40] S. Dakshanamurthy, N. T. Issa, S. Assefnia, A. Seshasayee, O. J. Peters, S. Madhavan, A. Uren, M. L. Brown y S. W. Byers, "Predicting New Indications for Approved Drugs Using a Proteochemometric Method," *Journal of Medicinal Chemistry*, volumen 55, número 15, páginas 6832-6848, 2012.

[41] S. Yuan, J. F.-W. Chan, H. Den-Haan, K. K.-H. Chik, A. J. Zhang, C. C.-S. Chan, V. K.-M. Poon, C. C.-Y. Yip, W. W.-N. Mak, Z. Zhu, Z. Zou, K.-M. Tee, J.-P. Cai, K.-H. Chan, J. de la Peña, H. Pérez-Sánchez, J. P. Cerón-Carrasco y K.-Y. Yuen, "Structure-based discovery of clinically approved drugs as Zika virus NS2B-NS3 protease inhibitors that potently inhibit Zika virus infection in vitro and in vivo," *Antiviral Research*, volumen 145, páginas 33-43, 2017.

[42] D. A. Case, D. S. Cerutti, T. E. Cheatham, T. A. Darden, R. E. Duke, T. J. Giese, H. Gohlke, A. W. Goetz, D. Greene, N. Homeyer, S. Izadi, A. Kovalenko, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D. Mermelstein, K. M. Merz, G. Monard, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D. R. Roe, A. Roitberg, C. Sagui, C. L. Simmerling, W. M. Botello-Smith, J. Swails, R. C. Walker, J. Wang, R. M. Wolf, X. Wu, L. Xiao, D. M. York y P. A. Kollman, "AMBER," *San Francisco: University of California*, 2017.

[43] T. A. Halgren, "Potential energy functions," *Current Opinion in Structural Biology*, volumen 5, número 2, páginas 205-210, 1995.

[44] *Maybridge,* `https://www.maybridge.com/`, Último acceso: 2018-09-09.

[45] D. Butina, "Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets," *Journal of Chemical Information and Computer Sciences*, volumen 39, número 4, páginas 747-750, 1999.

[46] C. A. Lipinski, F. Lombardo, B. W. Dominy y P. J. Feeney, "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings1PII of original article: S0169-409X(96)00423-1. The article was originally published in Advanced Drug Delivery Reviews 23 (1997) 3," *Advanced Drug Delivery Reviews*, volumen 46, número 1, páginas 3-26, 2001.

[47] S. Balani, G. Miwa, L.-S. Gan, J.-T. Wu y F. Lee, "Strategy of Utilizing In Vitro and In Vivo ADME Tools for Lead Optimization and Drug Candidate Selection," *Current Topics in Medicinal Chemistry*, volumen 5, número 11, páginas 1033-1038, 2005.

[48] N. Huang, B. K. Shoichet y J. J. Irwin, "Benchmarking Sets for Molecular Docking," *Journal of Medicinal Chemistry*, volumen 49, número 23, páginas 6789-6801, 2006.

[49] W. D. Ihlenfeldt, Y. Takahashi, H. Abe y S.-i. Sasaki, "Computation and management of chemical properties in CACTVS: An extensible networked approach toward modularity and compatibility," *Journal of chemical information and computer sciences*, volumen 34, número 1, páginas 109-116, 1994.

[50] M. M. Mysinger, M. Carchia, J. J. Irwin y B. K. Shoichet, "Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking," *Journal of Medicinal Chemistry*, volumen 55, número 14, páginas 6582-6594, 2012.

[51] I. Wallach, M. Dzamba y A. Heifets, "AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery," *arXiv preprint arXiv:1510.02855*, 2015.

[52] K. Deb, K. Sindhya y J. Hakanen, "Multi-objective optimization," en, CRC Press, 2016, páginas 145-184.

[53] E. Zitzler, J. Knowles y L. Thiele, "Quality assessment of Pareto set approximations," en, volumen 5252, Springer. Lecture Notes in Computer Science, 2008, páginas 373-404.

[54] L. While, L. Bradstreet y L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, volumen 16, número 1, páginas 86-95, 2012.

[55] E. Zitzler y L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study," Springer-Verlag, 1998, páginas 292-301.

[56] E. K. Burke y G Kendall, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science & Business Media, 2013.

[57] C. A. C. Coello, D. A. Van Veldhuizen, G. B. Lamont y D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, volumen 242.

[58] F. J. Solis y R. J.-B. Wets, "Minimization by Random Search Techniques," *Mathematics of Operations Research*, volumen 6, número 1, páginas 19-30, 1981.

[59] A. Lancinskas, P. M. Ortigosa y J. Zilinskas, "Multi-objective single agent stochastic search in non-dominated sorting genetic algorithm," *Nonlinear Analysis: Modelling and Control*, volumen 18, número 3, páginas 293-313, 2013.

[60] J. L. Redondo, *Solving competitive location problems via memetic algorithms. High performance computing approaches*. Universidad de Almería, 2009.

[61] E. Alba, *Parallel metaheuristics: a new class of algorithms*. John Wiley & Sons, 2005, volumen 47.

[62] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, volumen 13, número 5, páginas 533-549, 1986.

[63] C. Blum y A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, volumen 35, número 3, páginas 189-213, 2003.

[64] M. Birattari, L. Paquete, T. Stützle y K. Varrentrapp, "Classification of metaheuristics and design of experiments for the analysis of components," *Teknik Rapor, AIDA-01-05*, 2001.

[65] M. Dorigo, "Ant colony optimization," *Scholarpedia*, volumen 2, número 3, página 1461, 2007.

[66] M. Dorigo y G. Di Caro, "New Ideas in Optimization," en, McGraw-Hill Ltd., UK, 1999, páginas 11-32.

[67] L. Yu, K. Liu y K. Li, "Ant colony optimization in continuous problem," *Frontiers of Mechanical Engineering in China*, volumen 2, número 4, páginas 459-462, 2007.

[68] R. Kicinger, T. Arciszewski y K. D. Jong, "Evolutionary Computation and Structural Design: A Survey of the State-of-the-art," *Comput. Struct.*, volumen 83, número 23-24, páginas 1943-1978, 2005.

[69] S. Luke, *Issues in scaling genetic programming: breeding strategies, tree generation, and code bloat*, 2000.

[70] C. Darwin, "The origin of species by means of natural selection," 1859.

[71] J. H. Holland, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[72] A. E. Eiben, P.-E. Raue y Z. Ruttkay, "Genetic algorithms with multi-parent recombination," Springer, 1994, páginas 78-87.

[73] H. Bersini, "In search of a good evolution-optimization crossover," *Parallel Problem Solving form Nature*, 1992.

[74] H. Muhlenbein y H. Voigt, "Gene pool recombination in genetic algorithms," volumen 1, Kluwer Academic Publishers, 1995, páginas 53-62.

[75] G. Syswerda, "Simulated crossover in genetic algorithms," en, volumen 2, Elsevier, 1993, páginas 239-255.

[76] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems.," Tesis doctoral, 1975.

[77] S. W. Mahfoud, "Niching methods for genetic algorithms," *Urbana*, volumen 51, número 95001, páginas 62-94, 1995.

[78] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[79] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, 1994.

[80] A. E. Eiben y J. E. Smith, *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[81] H.-G. Beyer y H.-P. Schwefel, "Evolution Strategies &Ndash;A Comprehensive Introduction," volumen 1, número 1, páginas 3-52, 2002.

[82] *Learning Classifier Systems: From Foundations to Applications*. Springer, Berlin, Heidelberg, 2000, volumen 1813.

[83] P. Moscato, "On genetic crossover operators for relative order preservation," *C3P Report*, volumen 778, 1989.

[84] P. Moscato y C. Cotta, "Memetic Algorithms," 2005.

[85] C. Cotta-Porras, "A Study of Hybridisation Techniques and Their Application to the Design of Evolutionary Algorithms," *AI Commun.*, volumen 11, número 3,4, páginas 223-224, 1998.

[86] *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[87] D. H. Wolpert y W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, volumen 1, número 1, páginas 67-82, 1997.

[88] R. Dawkins, "The selfish gene New York: Oxford University Press," *DawkinsThe Selfish Gene1976*, 1976.

[89] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, volumen 826, página 1989, 1989.

[90] P. Moscato, C. Cotta y A. Mendes, "Memetic Algorithms," en, 2004, páginas 53-85.

[91] S. J. Louis, X. Yin y Z. Y. Yuan, "Multiple vehicle routing with time windows using genetic algorithms," volumen 3, IEEE, 1999, páginas 1804-1808.

[92] P. D. Surry y N. J. Radcliffe, "Inoculation to initialise evolutionary search," Springer, 1996, páginas 269-285.

[93] P. Moscato y C. Cotta, "A Gentle Introduction to Memetic Algorithms," en, Kluwer Academic Publishers, 2003, páginas 105-144.

[94] P. Moscato y C. Cotta Porras, "An Introduction to Memetic Algorithms," *Inteligencia Artificial*, volumen 7, número 19, 2003.

[95] M. Flynn, "Some computer organizations and their effectiveness," *IEEE Transactions on Computers*, volumen 21, número 9, páginas 948-960, 1972.

[96] D. A. Patterson y J. L. Hennessy, *Computer Organization and Design ARM Edition: The Hardware Software Interface*. Morgan Kaufmann, 2016.

[97] J. L. Hennessy y D. A. Patterson, *Computer architecture, fourth edition: A quantitative approach*. Morgan Kaufmann Publishers Inc., 2006.

[98] B. Nichols, D. Buttlar, J. Farrell y J. Farrell, *Pthreads programming: A POSIX standard for better multiprocessing*. O'Reilly Media, Inc., 1996.

[99] OpenMP. (2018). The OpenMP API specification for parallel programming.

[100] G. Alliance. (2018). Globus Toolkit.

[101] A. B. Yoo, M. A. Jette y M. Grondona, "SLURM: Simple Linux Utility for Resource Management," en, 2003, páginas 44-60.

[102] Universidad de Chile, *Laboratorio Nacional de Computación de Alto Rendimiento*, `https://www.nlhpc.cl/`, Último acceso: 2020-09-01.

[103] Universidad de Málaga, *Supercomputing and Bioinnovation Center*, `http://www.scbi.uma.es/`, Último acceso: 2018-05-01.

[104] The Norwegian e-infrastructure for Research Education, *Stallo*, `http://hpc.uit.no`, Último acceso: 2020-09-1.

[105] Institute of Bioorganic Chemistry of the Polish Academy of Sciences, *Poznan Supercomputing and Networking Center*, `https://www.psnc.pl/`, Último acceso: 2020-09-11.

[106] Supercomputación y Algoritmos, *Bullxual*, `https://hpca.ual.es/`, Último acceso: 2020-09-16.

[107] S. Puertas-Martín, J. L. Redondo, P. M. Ortigosa y H. Pérez-Sánchez, "OptiPharm: An evolutionary algorithm to compare shape similarity," *Scientific Reports*, volumen 9, número 1, página 1398, 2019.

[108] W. R. Hamilton, "Theory of Quaternions," *Proceedings of the Royal Irish Academy (1836-1869)*, volumen 3, páginas 1-16, 1844.

[109] M. Jelásity, P. M. Ortigosa e I. García, "UEGO, An Abstract Clustering Technique for Multimodal Global Optimization," *Journal of Heuristics*, volumen 7, número 3, páginas 215-233, 2001.

[110] P. M. Ortigosa, I. García y M. Jelásity, "Reliability and performance of UEGO , a clustering-based global optimizer," *Journal of Global Optimization*, volumen 19, número 3, páginas 265-289, 2001.

[111] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, volumen 2, número 11, páginas 559-572, 1901.

[112] OpenEye Scientific Software, "ROCS," *Santa Fe, NM. www.eyesopen.com*, 2019.

[113] J. L. Redondo, J. Fernández, I. García y P. M. Ortigosa, "Solving the Multiple Competitive Location and Design Problem on the Plane," *Evolutionary Computation*, volumen 17, número 1, páginas 21-53, 2009.

[114] J. L. Redondo, P. M. Ortigosa y J. Zilinskas, "Multimodal evolutionary algorithm for multidimensional scaling with city-block distances," *Informatica*, volumen 23, número 4, páginas 601-620, 2012.

[115] M. E. H. Petering y M. I. Hussein, "A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem," *European Journal of Operational Research*, volumen 231, número 1, páginas 120-130, 2013.

[116] B. Ivorra, M. R. Ferrández, M. Crespo, J. L. Redondo, P. M. Ortigosa, J. G. Santiago y Á. M. Ramos, "Modelling and optimization applied to the design of fast hydrodynamic focusing microfluidic mixer for protein folding," *Journal of Mathematics in Industry*, volumen 8, número 1, página 4, 2018.

[117] J. Fernández, B. G. Tóth, J. L. Redondo y P. M. Ortigosa, "The probabilistic customer's choice rule with a threshold attraction value: Effect on the location of competitive facilities in the plane," *Computers and Operations Research*, volumen 101, páginas 234-249, 2019.

[118] R. A. Johnson y G. K. Bhattacharyya, *Statistics: Principles and Methods*. John Wiley & Sons, 2014, página 736.

[119] H. Ge, Y. Wang, W. Zhao, W. Lin, X. Yan y J. Xu, "Saffold hopping of potential anti-tumor agents by WEGA: a shape-based approach," *MedChemComm*, volumen 5, número 6, páginas 737-741, 2014.

[120] S. Lešnik, T. Štular, B. Brus, D. Knez, S. Gobec, D. Janežič y J. Konc, "LiSiCA: A Software for Ligand-Based Virtual Screening and Its Application for the Discovery of Butyrylcholinesterase Inhibitors," *Journal of Chemical Information and Modeling*, volumen 55, número 8, páginas 1521-1528, 2015.

[121] G. Tresadern, D. Bemporad y T. Howe, "A comparison of ligand based virtual screening methods and application to corticotropin releasing factor 1 receptor," *Journal of Molecular Graphics and Modelling*, volumen 27, número 8, páginas 860-870, 2009.

[122] S. Chu y M. Gochin, "Identification of fragments targeting an alternative pocket on HIV-1 gp41 by NMR screening and similarity searching," *Bioorganic and Medicinal Chemistry Letters*, volumen 23, número 18, páginas 5114-5118, 2013.

[123] E.-S. Kim, H. Cho, C. Lim, J.-Y. Lee, D.-I. Lee, S. Kim y A. Moon, "A natural piperamide-like compound NED-135 exhibits a potent inhibitory effect on the invasive breast cancer cells," *Chemico-Biological Interactions*, volumen 237, páginas 58-65, 2015.

[124] B. R. Kossmann, M. Abdelmalak, S. Lopez, G. Tender, C. Yan, Y. Pommier, C. Marchand e I. Ivanov, "Discovery of selective inhibitors of tyrosyl-DNA phosphodiesterase 2 by targeting the enzyme DNA-binding cleft," *Bioorganic and Medicinal Chemistry Letters*, volumen 26, número 14, páginas 3232-3236, 2016.

[125] J. L. Woodring, K. A. Bachovchin, K. G. Brady, M. F. Gallerstein, J. Erath, S. Tanghe, S. E. Leed, A. Rodriguez, K. Mensa-Wilmot, R. J. Sciotti y M. P. Pollastri, "Optimization of physicochemical properties for 4-anilinoquinazoline inhibitors of trypanosome proliferation," *European Journal of Medicinal Chemistry*, volumen 141, páginas 446-459, 2017.

[126] G. MacCari, T. Jaeger, F. Moraca, M. Biava, L. Flohé y M. Botta, "A fast virtual screening approach to identify structurally diverse inhibitors of trypanothione reductase," *Bioorganic and Medicinal Chemistry Letters*, volumen 21, número 18, páginas 5255-5258, 2011.

[127] Y.-R. Kim, H.-J. Koh, J.-S. Kim, J.-S. Yun, K. Jang, J.-Y. Lee, J. U. Jung y C.-S. Yang, "Peptide inhibition of p22phox and Rubicon interaction as a therapeutic strategy for septic shock," *Biomaterials*, volumen 101, páginas 47-59, 2016.

[128] M. López-Ramos y F. Perruccio, "HPPD: Ligand- and Target-Based Virtual Screening on a Herbicide Target," *Journal of Chemical Information and Modeling*, volumen 50, número 5, páginas 801-814, 2010.

[129] K. E. Hevener, S. Mehboob, P.-C. Su, K. Truong, T. Boci, J. Deng, M. Ghassemi, J. L. Cook y M. E. Johnson, "Discovery of a Novel and Potent Class of F. tularensis Enoyl-Reductase (FabI) Inhibitors by Molecular Shape and Electrostatic Matching," *Journal of Medicinal Chemistry*, volumen 55, número 1, páginas 268-279, 2012.

[130] T. S. Kaoud, C. Yan, S. Mitra, C.-C. Tseng, J. Jose, J. M. Taliaferro, M. Tuohetahuntila, A. Devkota, R. Sammons, J. Park, H. Park, Y. Shi, J. Hong, P. Ren y K. N. Dalby, "From in Silico Discovery to Intracellular Activity: Targeting JNK–Protein Interactions with Small Molecules," *ACS Medicinal Chemistry Letters*, volumen 3, número 9, páginas 721-725, 2012.

[131] P. Tiikkainen, P. Markt, G. Wolber, J. Kirchmair, S. Distinto, A. Poso y O. Kallioniemi, "Critical Comparison of Virtual Screening Methods against the MUV Data Set," *Journal of Chemical Information and Modeling*, volumen 49, número 10, páginas 2168-2178, 2009.

[132] A. Massarotti, A. Brunco, G. Sorba y G. C. Tron, "ZINClick: A Database of 16 Million Novel, Patentable, and Readily Synthesizable 1,4-Disubstituted Triazoles," *Journal of Chemical Information and Modeling*, volumen 54, número 2, páginas 396-406, 2014.

[133] J. Oyarzabal, T. Howe, J. Alcazar, J. I. Andrés, R. M. Alvarez, F. Dautzenberg, L. Iturrino, S. Martınez e I. Van der Linden, "Novel Approach for Chemotype Hopping Based on Annotated Databases of Chemically Feasible Fragments and a Prospective Case Study: New Melanin Concentrating Hormone Antagonists," *Journal of Medicinal Chemistry*, volumen 52, número 7, páginas 2076-2089, 2009.

[134] OpenEye Scientific Software, "Zap Toolkit," *Santa Fe, NM. www.eyesopen.com*, 2019.

[135] B. A. Ellingson, A. G. Skillman y A. Nicholls, "A nalysis of S M8 and Z ap T K calculations and their geometric sensitivity," *Journal of Computer-Aided Molecular Design*, volumen 24, número 4, páginas 335-342, 2010.

[136] D. G. Thomas, J. Chun, Z. Chen, G. Wei y N. A. Baker, "Parameterization of a geometric flow implicit solvation model," *Journal of Computational Chemistry*, volumen 34, número 8, páginas 687-695, 2013.

[137] P. C. D. Hawkins y G. Stahl, "Ligand-Based Methods in GPCR Computer-Aided Drug Design," en, 2018, páginas 365-374.

[138] P. R. Connelly, P. W. Snyder, Y. Zhang, B. McClain, B. P. Quinn, S. Johnston, A. Medek, J. Tanoury, J. Griffith, W Patrick Walters, E. Dokou, D. Knezic y P. Bransford, "The potency−insolubility conundrum in pharmaceuticals: Mechanism and solution for hepatitis C protease inhibitors," *Biophysical Chemistry*, volumen 196, páginas 100-108, 2015.

[139] R. Gowthaman, S. Lyskov y J. Karanicolas, "DARC 2.0: Improved Docking and Virtual Screening at Protein Interaction Sites," *PLOS ONE*, volumen 10, número 7, e0131612, 2015.

[140] J. J. Durillo y A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, volumen 42, páginas 760-771, 2011.

[141] S. Kukkonen y J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings*, volumen 1, páginas 443-450, 2005.

[142] E. Zitzler y S. Künzli, "Indicator-based selection in multiobjective search," 2004, páginas 832-842.

[143] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro y E. Alba, "MOCell: A cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, volumen 24, número 7, páginas 726-746, 2009.

[144] Q. Zhang y H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, volumen 11, número 6, páginas 712-731, 2007.

[145] K. Deb, A. Pratap, S. Agarwal y T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, volumen 6, número 2, páginas 182-197, 2002.

[146] M. R. Sierra y C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and -dominance," 2005, páginas 505-519.

[147] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna y E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," 2009, páginas 66-73.

[148] E. Zitzler, M. Laumanns y L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-report*, volumen 103, 2001.

[149] A. J. Banegas-Luna, J. P. Cerón-Carrasco, S. Puertas-Martín y H. Pérez-Sánchez, "BRUSELAS: HPC Generic and Customizable Software Architecture for 3D Ligand-Based Virtual Screening of Large Molecular Databases," *Journal of Chemical Information and Modeling*, volumen 59, número 6, páginas 2805-2817, 2019.

[150] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, volumen 27, número 8, páginas 861-874, 2006.

[151] J. P. Egan, *Signal detection theory and ROC-analysis*. New York : Academic Press, 1975.

[152] J. A. Swets, R. M. Dawes y J Monahan, "Better decisions through science.," *Scientific American*, volumen 283, número 4, páginas 82-87, 2000.

[153] K. H. Zou, "Receiver operating characteristic (ROC) literature research," 2002.

[154] K. A. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," 1989, páginas 160-163.

[155] J. Sebaugh, "Guidelines for accurate EC50/IC50 estimation," *Pharmaceutical statistics*, volumen 10, número 2, páginas 128-134, 2011.

[156] K. Pearson, *Mathematical contributions to the theory of evolution*, v. 13-17. Dulau y co., 1904.

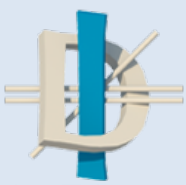
[157] J. Boström, J. A. Grant, O. Fjellström, A. Thelin y D. Gustafsson, "Potent Fibrinolysis Inhibitor Discovered by Shape and Electrostatic Complementarity to the Drug Tranexamic Acid," *Journal of Medicinal Chemistry*, volumen 56, número 8, páginas 3273-3280, 2013.

[158] I. Haque y V. Pande, "Method for rapidly approximating similarities. Patent number: US8706427B2," patente US8706427B2.

[159] OpenEye Scientific Software, "VIDA 4.4.0.4," *Santa Fe, NM. www.eyesopen.com*, 2019.

[160] A. J. Banegas-Luna, J. P. Cerón-Carrasco, S. Puertas-Martín y H. Pérez-Sánchez, "BRUSELAS: HPC Generic and Customizable Software Architecture for 3D Ligand-Based Virtual Screening of Large Molecular Databases," *Journal of Chemical Information and Modeling*, volumen 59, número 6, páginas 2805-2817, 2019.

UNIVERSIDAD DE ALMERÍA



DEPARTAMENTO
DE INFORMÁTICA