Universitat Oberta de Catalunya

UNIVERSITAT DE BARCELONA

Metodologías de última generación para el modelado estructural y energético de complejos entre proteínas

Máster Universitario en Bioinformática y Bioestadística

Trabajo Fin Máster

Area 2: Drug Design and Structural Biology

Tutor: Juan Fernández Recio

Alumno: Miguel Ángel Jiménez Carrasco

15/enero/2023

Título del trabajo:	*Metodologías de última generación para el modelado estructural y energético de complejos entre proteínas*
Nombre del autor:	*Miguel Ángel Jiménez Carrasco*
Nombre del consultor/a:	*Juan Fernández Recio*
Nombre del PRA:	*Nuria Pérez Álvarez*
Fecha de entrega (mm/aaaa):	*01/2023*
Titulación o programa:	Máster Universitario en Bioinformática y Bioestadística
Área del Trabajo Final:	*Area 2: Drug Design and Structural Biology*
Idioma del trabajo:	*Español*
Palabras clave	*Protein-protein interactions, docking, affinity, composite scoring functions, structural biology, Machine Learning bioinformatics*

Resumen del Trabajo

En este trabajo se exploran nuevos protocolos para abordar el modelado de interacciones en sistemas de interés biológico y biotecnológico. Se parte de un conjunto de complejos entre proteínas que disponen de estructuras de alta resolución tanto para los complejos como para sus componentes no unidos, y cuyas constantes de disociación se han medido por métodos biofísicos. Las interacciones no covalentes que muchas proteínas establecen con otras biomoléculas son esenciales para su función. Cuando se dispone de una estructura de alta resolución del complejo, la interacción puede describirse a detalle atómico, pero es la afinidad de los componentes entre sí lo que determina la fortaleza del ensamblaje en unas condiciones dadas de temperatura, pH y concentración de proteínas, y si éste es transitorio o permanente. Varios modelos han intentado correlacionar la afinidad con las características observadas en estas estructuras. Aunque algunos han tenido mucho éxito en conjuntos de entrenamiento pequeños, los modelos publicados no han funcionado tan bien en conjuntos más grandes, y su valor predictivo sigue siendo pobre. El trabajo se centrará en explorar metodologías de última generación para el modelado estructural y energético de complejos entre proteínas. En concreto, se explora el uso de algoritmos de Machine Learning para la clasificación y predicción de la afinidad experimental a partir de los valores calculados de potencial de CCharPPI

Abstract

This work explores new protocols for modelling interactions in systems of biological and biotechnological interest. The starting point is a set of protein-protein complexes that have high-resolution structures for both the complexes and their unbound components, and whose dissociation constants have been measured by biophysical methods. The non-covalent

interactions that many proteins establish with other biomolecules are essential for their function. When a high-resolution structure of the complex is available, the interaction can be described in atomic detail, but the experimentally determined binding affinity will define the strength of the assembly under given conditions of temperature, pH and protein concentration, and whether it is transient or permanent. Several models have attempted to correlate binding affinity with the features observed in these structures. While some have been very successful on small training sets, published models have not performed as well on larger sets, and their predictive value remains poor. The work will focus on exploring state-of-the-art methodologies for structural and energetic modelling of protein-protein complexes. In particular, the use of Machine Learning algorithms for classification and prediction of experimental affinity from calculated CCharPPI potential values is explored

Contenido

1. Introducción 1
1.1. Contexto y justificación del Trabajo 1
Importancia de las proteínas en los organismos vivos 1
Complejos entre proteínas 1
Determinación y predicción de estructura 2
Determinación y predicción de la energía de unión 2
Antecedentes / Motivación 2
Limitaciones actuales a resolver 2
Impacto 3
1.2. Objetivos del Trabajo 3
Objetivos generales 3
Objetivos específicos 3
1.3. Impacto en sostenibilidad, ético-social y de diversidad 3
1.4. Enfoque y método seguido 4
Enfoques posibles de partida 4
Enfoque escogido 4
Machine Learning (ML) 4
Funciones de modelado energético 4
1.5. Planificación del Trabajo 5
Tareas 6
Calendario 7
Hitos 7
Análisis de riesgos 7
1.6. Breve sumario de productos obtenidos 8
Resultados esperados 8
1.7. Breve descripción de los otros capítulos de la memoria 8
2. Estado del arte 10
2.1. Visión retrospectiva sobre afinidad y algoritmos de predicción [1,2] 10
2.2. Funciones de caracterización energética de complejos. Servidor CCharPPI [3] 13
2.3. Aprendizaje automático 14
¿Qué es el aprendizaje automático? 14
Tipos de sistemas basados en aprendizaje automático 14
Puntos a tener en cuenta a la hora de su aplicación 15
Entrenamiento y validación 15

Pasos para la creación de modelos basados en aprendizaje automático 16
2.4. Metodologías de última generación para el modelado estructural y energético de complejos entre proteínas 17
3. Materiales y métodos 19
3.1. Entornos de trabajo 19
3.2. Datos de partida 19
3.3. Carga y limpieza de datos 19
3.4. Datos de trabajo en SQL Server 20
4. Resultados 21
4.1. Cálculos CCharPPI 21
4.2. Análisis de correlaciones 22
4.3. Análisis de clasificación 25
4.4. Definición de nuevos métodos y técnicas a usar 28
4.5. Desarrollo del modelo 28
Modelos basados en K-means 28
Modelos basados en Random Forest 32
PCA como herramienta para reducción de dimensiones 32
5. Conclusiones y trabajos futuros 34
5.1. Conclusiones 34
5.2. Trabajos futuros 34
6. Glosario 35
7. Bibliografía 36
7.1. Documentos 36
URLs 36
8. Anexos 37
8.1. Repositorio en GitHub 37
8.2. Algunos ejemplos de código Python 37

Lista de figuras

Figura 1: Complejo entre proteínas [Molecule of the Month. Fuente: https://pdb101.rcsb.org/motm/274] 1
Figura 2: Tipos de algoritmos de ML [Adaptado de: Lantz (2019)] 4
Figura 3: Servidor CCharPPI 5
Figura 4: Calendario PECs 6
Figura 5: Calendario TFM 7
Figura 6: Evaluación de los métodos 12
Figura 7: Algoritmos de ML (Lantz) 15
Figura 8: Pasos en ML (Géron) 16
Figura 9: Importación de Excel desde PDF 20
Figura 10: Estructuras SQL Server 20
Figura 11: Estudio de valores nan 22
Figura 12: Correlación con ΔG 23
Figura 13: Correlación 2 a 2 23
Figura 14: Ejemplo de correlación baja 24
Figura 15: Ejemplo de correlación alta 24
Figura 16: Análisis de clasificación 25
Figura 17: Función de densidad por grupos 26
Figura 18: Histograma por grupos 27
Figura 19: Distribución de afinidad según descriptores (1) 27
Figura 20: Distribución de afinidad según descriptores (2) 28
Figura 21: Cruce de descriptores 29
Figura 22: Puntos de partida 30
Figura 23: Curva para determinar K 30
Figura 24: Agrupaciones K-means 31
Figura 25: Proyecciones 2D 32
Figura 26: Variabilidad acumulada 33
Figura 27: Visión reducida a dos componentes 33

1. Introducción

1.1. Contexto y justificación del Trabajo

Importancia de las proteínas en los organismos vivos

Las proteínas son elementos indispensables en los procesos químicos de las células. Intervienen en aspectos tan importantes como la catálisis de procesos metabólicos, en la transferencia de energía, en la respiración, en la fotosíntesis, en la expresión genética, en el transporte a través de las membranas, en la comunicación celular, en el reconocimiento molecular,…

Las proteínas se constituyen como cadenas de aminoácidos. La secuencia de aminoácidos constituye su estructura primaria. Determinados segmentos tienden a plegarse en formas simples como hélices, giros,… Es lo que constituye la estructura secundaria. La cadena completa se estabiliza y cumple su función al plegarse en una estructura tridimensional compacta. Se conforma bajo su estructura terciaria. En un nivel superior, estructura cuaternaria, varias cadenas se unen para constituir un complejo.

Complejos entre proteínas

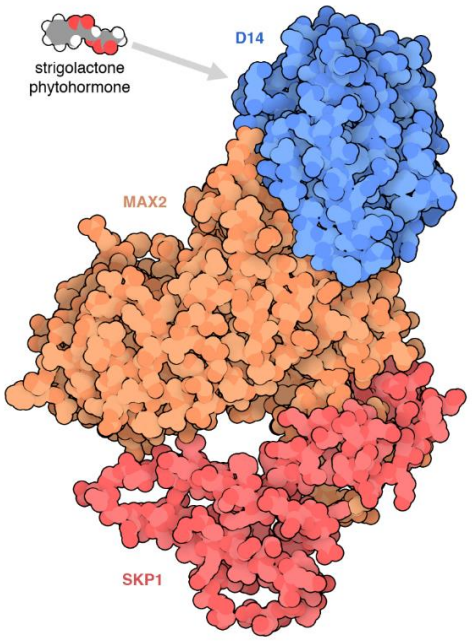

Figura 1: Complejo entre proteínas [Molecule of the Month. Fuente: https://pdb101.rcsb.org/motm/274]

Las funciones de las proteínas se basan en su complejidad estructural. Esta complejidad estructural permite, además, la formación de complejos donde se acoplan con otras moléculas (ligandos). Tales ligandos pueden ser moléculas pequeñas o grandes. En este último tipo podemos englobar los complejos entre proteínas. Los complejos pueden intervenir en funciones tales como:

- Catálisis
- Regulación de la actividad
- Comunicación
- Agente defensivo o atacante

La unión de los elementos del complejo tiene distintos aspectos. Existe un aspecto geométrico, de encaje entre las formas. Además, existe un componente energético entre los

átomos a considerar. En conjunto, los aspectos estructurales, termodinámicos y cinéticos de la interacción entre dos proteínas resultan fundamentales para conocer mejor su mecanismo y función, entender sus implicaciones biológicas y biomédicas (diagnóstico y tratamiento de enfermedades), así como para poder modular dicha interacción en casos de interés biotecnológico (ingeniería y diseño de proteínas, vacunas o biosensores).

Determinación y predicción de estructura

Se utilizan técnicas experimentales para conocer la estructura 3D. Entre ellas, podemos citar algunas:

- Difracción de rayos X
- Espectroscopia de resonancia magnética nuclear (RMN)
- Microscopía crío-electrónica

En los casos en los que no se ha determinado experimentalmente la estructura de una biomolécula, resultan útiles las técnicas de predicción estructural, mediante el uso de la bioinformática.

Determinación y predicción de la energía de unión

Una descripción computacional más completa de la interacción proteína-proteína requiere también del conocimiento de las afinidades de unión.

En los casos en los que no se ha determinado experimentalmente, resultan, asimismo, útiles los algoritmos que puedan predecir estas afinidades de unión. Aunque las funciones de energía para la predicción de la afinidad y la clasificación de las posturas de acoplamiento están relacionadas, a menudo se desarrollan específicamente para sus respectivos fines y hasta ahora han mostrado un rendimiento variable y bastante limitado.

Antecedentes / Motivación

Actualmente, se dispone de la estructura 3D determinada experimentalmente solo para una pequeña fracción de todos los posibles complejos existentes entre proteínas humanas (y en otros organismos).

Para complementar la falta de datos experimentales, en estos últimos años se han producido avances significativos en el modelado molecular de complejos entre proteínas, en base a métodos de modelado por homología (solo en casos en los que se disponga estructura de un complejo formado entre proteínas homólogas a las estudiadas), docking y dinámica molecular, y más recientemente, mediante metodologías basadas en inteligencia artificial (IA), como AF-Multimer, con resultados predictivos prometedores como se ha puesto en evidencia en los experimentos de evaluación de metodologías CAPRI y CASP.

Limitaciones actuales a resolver

A pesar de los avances, aún quedan retos importantes por resolver. Como se ha puesto de manifiesto en el último experimento CASP15 recientemente evaluado (https://predictioncenter.org/casp15/zscores_multimer.cgi), los métodos basados en IA aplicados de forma automática no proporcionan predicciones tan fiables como cuando se integran con otras metodologías más tradicionales. En general, estos métodos tienen dificultades para modelar complejos con anticuerpos, no permiten describir procesos dinámicos, y no proporcionan datos sobre la energética de la interacción o el impacto de mutaciones. En este sentido, aun disponiendo de la estructura tridimensional de un complejo entre proteínas (determinada de forma experimental o modelada), no resulta obvio estimar la energía de unión o afinidad de dicho complejo, y se necesitan nuevas metodologías con una capacidad predictiva más fiable.

Impacto

El desarrollo de nuevos protocolos para abordar dichos retos permitiría su aplicación al modelado de interacciones en sistemas de interés biológico y biotecnológico, con aplicaciones futuras como:

- Diseño de fármacos contra interacciones específicas
- Interpretación y predicción de mutaciones patológicas
- Caracterización de variedades genéticas en diferentes organismos
- Mejora de levaduras y otros microorganismos de interés industrial,
- Diseño de biopesticidas y compuestos antifúngicos
- Diseño de anticuerpos terapéuticos y optimización de vacunas

1.2. Objetivos del Trabajo

Objetivos generales

El objetivo general es el desarrollo de metodologías de última generación para la estimación de la energía de unión de complejos entre proteínas a partir de su estructura 3D, en base a métodos de optimización multiparamétrica mediante aprendizaje automático.

Objetivos específicos

Se aplicarán funciones de puntuación de pyDock, ConsSurf y las funciones de CCharPPI, para mejorar la predicción de afinidad de complejos. Se aplicarán técnicas de optimización y aprendizaje automático, para encontrar las mejores combinaciones entre las funciones mencionadas.

Se integrará lo generado por el trabajo dentro de un marco metodológico general que cubra el modelado estructural y energético de complejos entre proteínas

1.3. Impacto en sostenibilidad, ético-social y de diversidad

Las proteínas son elementos indispensables en los procesos químicos de las células. Intervienen en aspectos tan importantes como la catálisis de procesos metabólicos, en la transferencia de energía, en la respiración, en la fotosíntesis, en la expresión genética, en el transporte a través de las membranas, en la comunicación celular, en el reconocimiento molecular,…

Como en el resto de las áreas que involucran el progreso y la investigación científica, se deben contemplar tres grandes dimensiones:

- Sostenibilidad,
- comportamiento ético y responsabilidad social, y
- diversidad y derechos humanos.

El desarrollo de protocolos de predicción basados en computación permite reducir los costes de determinación y la energía necesarios por la vía experimental. Esta reducción de coste permite abordar con mayor rapidez el modelado de interacciones en sistemas de interés biológico y biotecnológico.

Los ODS (Objetivos de Desarrollo Sostenible 2030, ONU), también deben guiar el foco de aplicación de las nuevas técnicas a problemas que tengan en cuenta aspectos como la reducción de la pobreza, la mejora de las condiciones de vida de la humanidad, la reducción de desigualdad y favoreciendo su aplicación, sin distinción de género, raza,…

Si bien el factor coste incide favorablemente en este foco, este elemento no es suficiente por sí sólo. No obstante, la incorporación de nuevos agentes de las áreas geográficas de mayor sensibilidad sí incidirá en dicho foco.

1.4. Enfoque y método seguido

Enfoques posibles de partida

Dentro del campo general del modelado estructural y energético de complejos entre proteínas, el trabajo se podría haber centrado en distintos aspectos.

Dentro del modelado estructural, con aproximaciones clásicas o con los nuevos enfoques basados en IA.

Dentro del modelado energético, con el cálculo de funciones o con el uso combinado de estos cálculos.

Enfoque escogido

Dados los avances que se han ido produciendo, en el campo de la predicción basada en IA, se identifica como un área de interés, el uso combinado de cálculos de funciones de modelado energético. Para este uso combinado, se usarán técnicas de optimización y aprendizaje automático.

Machine Learning (ML)

Se explorarán los distintos tipos de algoritmos para usar aquellos que produzcan mejor resultado en los objetivos definidos.

Dentro de los algoritmos, se explorará la construcción de modelos predictivos basados en aprendizaje supervisado, la clasificación en categorías, o la construcción de modelos descriptivos basados en aprendizaje no supervisado.

[Cuadro con tipos de algoritmos de Machine Learning. Fuente: Lantz (2019)]

Model	**Learning task**
Supervised learning algorithms	
k-nearest neighbors	Classification
Naive Bayes	Classification
Decision trees	Classification
Classification rule learners	Classification
Linear regression	Numeric prediction
Regression trees	Numeric prediction
Model trees	Numeric prediction
Neural networks	Dual use
Support vector machines	Dual use
Unsupervised learning algorithms	
Association rules	Pattern detection
k-means clustering	Clustering
Meta-learning algorithms	
Bagging	Dual use
Boosting	Dual use
Random forests	Dual use

Figura 2: Tipos de algoritmos de ML [Adaptado de: Lantz (2019)]

Funciones de modelado energético

Para el cálculo de las funciones de modelado energético, se utilizará el servidor CCharPPI (https://life.bsc.es/pid/ccharppi). Este servidor CCharPPI (Computational Characterisation of Protein-Protein Interactions) fue desarrollado por el grupo de investigación donde se ha llevado a cabo este trabajo (Moal et al 2015) y permite el cálculo de un gran número de descriptores

relacionados con aspectos energéticos de complejos entre proteínas a partir de las estructuras PDB de los componentes de estos complejos.

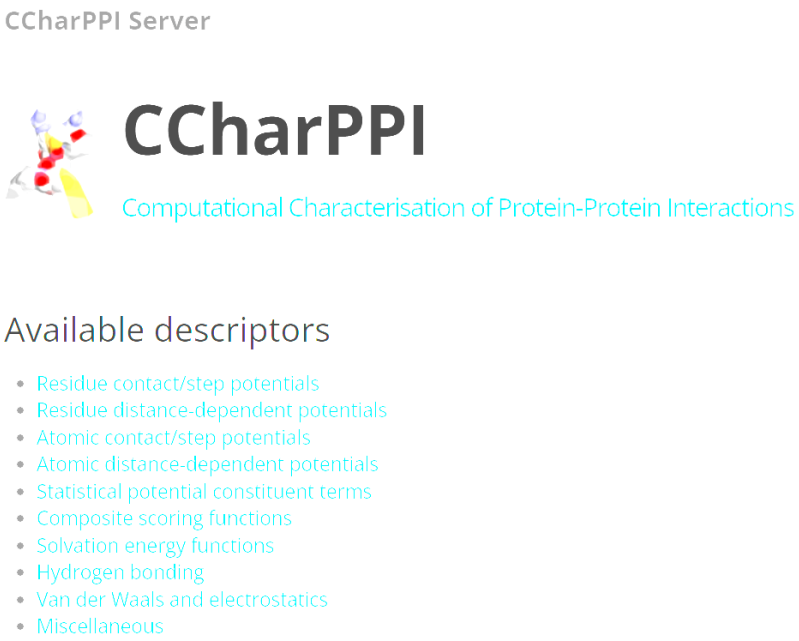

Figura 3: Servidor CCharPPI

1.5. Planificación del Trabajo

Se manejan dos perspectivas integradas de planificación de tareas: aquella que está orientada a cubrir los hitos académicos marcados por la UOC (PECs) y la derivada del desarrollo del proyecto técnico que cubre los objetivos del trabajo.

Las principales tareas y fechas a tener en cuenta desde la perspectiva académica son:

- PEC1: 17/10/2022
- PEC2 (Desarrollo Fase 1): 21/11/2022
- PEC3 (Desarrollo Fase 2): 24/12/2022
- PEC4: 15/01/2023
- PEC5: 03/02/2023

Tarea UOC	Septiembre	octubre				noviembre				diciembre					enero				febrero				
	28	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	13	20	27
PEC1				17																			
PEC2									21														
PEC3													24										
PEC4																15							
PEC5																			3				

Figura 4: Calendario PECs

Tareas

Integrando las tareas técnicas dentro de la perspectiva académica, llegamos a la siguiente lista:

- PEC1
  - Definición
  - Plan
- PEC2 (Desarrollo Fase 1)
  - Definición del resultado, datos de partida, formatos
  - Preparación de entornos de trabajo
  - Cálculos
  - Análisis de correlaciones
  - Definición de nuevos métodos y técnicas a usar
  - Desarrollo del modelo
  - Validación externa
- PEC3 (Desarrollo Fase 2)
  - Implementación del modelo (algoritmo)
  - Pruebas
  - Análisis de resultados y conclusiones
  - Generación del producto software
  - Futuros desarrollos (sujeto a disponibilidad de tiempo)
- PEC4
  - Memoria
  - Presentación
- PEC5
  - Defensa pública

Calendario

Partimos de las principales fechas a tener en cuenta desde la perspectiva académica y encajamos las tareas técnicas dentro de las fechas establecidas.

Para la planificación de tareas se tuvieron en cuenta los días disponibles para los periodos marcados, teniendo en cuenta todos los días (sin distinción) y una media de 3 horas por día. De esta manera, la planificación quedaría:

Tarea UOC	Subtarea UOC	Actividad técnica	Inicio	Fin	Días	Horas
PEC1			28/09/2022	17/10/2022	20	60
	Definición	Definición TFM				
	Plan	Planificación TFM				
PEC2			18/10/2022	21/11/2022	35	105
	Desarrollo Fase 1					
		Definición del resultado, datos de partida, formatos				
		Preparación de entornos de trabajo				
		Cálculos				
		Análisis de correlaciones				
		Definición de nuevos métodos y técnicas a usar				
		Desarrollo del modelo				
		Validación externa				
PEC3			22/11/2022	24/12/2022	33	99
	Desarrollo Fase 2					
		Implementación del modelo (algoritmo)				
		Pruebas				
		Análisis de resultados y conclusiones				
		Generación del producto software				
		Futuros desarrollos (sujeto a disponibilidad de tiempo)				
PEC4			25/12/2022	15/01/2023	22	66
	Memoria	Redacción de memoria				
	Presentación	Creación de presentación				
PEC5			16/01/2023	03/02/2023	19	57
	Defensa pública	Defensa del TFM				
Suma					129	387
			28/09/2022	03/02/2023	129	

Figura 5: Calendario TFM

Hitos

Siendo los principales hitos a tener en cuenta:

- PEC1: Documento de definición y plan
- PEC2 (Desarrollo Fase 1): Memoria
- PEC3 (Desarrollo Fase 2): Memoria y software
- PEC4: Memoria, software y presentación
- PEC5: Defensa

Análisis de riesgos

El principal riesgo de partida era que los algoritmos de ML no concluyeran con resultados satisfactorios. Para minimizar ese riesgo, fue importante abordar cuanto antes la obtención de datos de partida y la implementación de modelos para su análisis y validación.

Al planificar el trabajo, había otro riesgo importante a tener en cuenta. Los esfuerzos de las distintas actividades estaban muy ajustados. Ha sido importante, hacer un seguimiento continuo que identificara cualquier desviación. En caso de desviación, se tuvo que ajustar el alcance de los objetivos, ya que la fecha de finalización del TFM está marcada. Si bien fue posible algún ajuste entre las fases de desarrollo.

1.6. Breve sumario de productos obtenidos

Resultados esperados

Se obtuvieron los distintos productos durante el desarrollo del trabajo (definición y plan de trabajo) y como productos finales (memoria, software y presentación). En el caso de la memoria y el software, su configuración ha ido evolucionando hasta su versión final de TFM.

A continuación, se hace una breve descripción del contenido de cada producto.

Definición del trabajo

El documento define el TFM que se va a realizar.

Plan de trabajo

Se incluye una versión inicial del plan en el documento de definición. El plan se ha ido ajustando a medida que se ha ido desarrollando el trabajo. El plan refleja el estado actual en cada momento y sirve para hacer un seguimiento y analizar el impacto frente a desviaciones.

Memoria

La memoria del trabajo fin de máster constituye uno de los productos principales. Describe el objeto del trabajo, el desarrollo del trabajo realizado, los productos obtenidos y las conclusiones a las que se ha llegado.

Se crea bajo las recomendaciones de la plantilla 'TFMUBio_Plantilla_Memoria_es_v7_2022'.

Producto

Se establece como objetivo la construcción de software que aborde los objetivos marcados por el trabajo. Como software, consta de código, datos y documentación técnica y de usuario. Este software se integra dentro de un proceso para el modelado estructural y energético de complejos entre proteínas.

Se documentan casos de uso que describan la aplicación del proceso, haciendo especial énfasis en lo aportado por el TFM.

Presentación virtual

Para la presentación del trabajo, orientada a su defensa, se preparan los materiales de apoyo que ayudan en la explicación del trabajo realizado y los resultados obtenidos. Estos materiales sirven como material de apoyo en la defensa del TFM.

Se prioriza el uso de elementos visuales que faciliten la transmisión de lo realizado y obtenido por la realización del trabajo.

1.7. Breve descripción de los otros capítulos de la memoria

A este capítulo de introducción al trabajo, le siguen otros capítulos de la memoria.

En el capítulo sobre el estado del arte, se describe la situación actual de la materia del trabajo, que supone el punto de partida del trabajo desarrollado. En primer lugar, se realiza una visión retrospectiva sobre la afinidad y los algoritmos de predicción. Se describen, a continuación, las funciones de modelado energético. Se presenta el servidor CCharPPI que calcula un gran número de estos potenciales. Se pasa, entonces, a describir el aprendizaje automático como herramienta para la predicción de la afinidad. Por último, se establecen las

características de las metodologías de última generación que guían el modelado estructural y energético de complejos entre proteínas.

En el capítulo de materiales y métodos, se enumeran los aspectos más relevantes del diseño y desarrollo del trabajo, estableciendo la metodología elegida para realizar el desarrollo, describiendo las alternativas posibles, las decisiones tomadas, y los criterios utilizados para tomar estas decisiones. Además, se hace una descripción de los productos obtenidos.

En el capítulo de resultados, se detallan los resultados obtenidos en el trabajo, mediante la aplicación de los mencionados materiales y métodos.

En el capítulo de conclusiones y trabajos futuros, se hace una reflexión crítica sobre la consecución de los objetivos planteados. Se valora la planificación y metodología del trabajo. Y por último, se aventuran las posibles líneas de trabajo futuro que podrían ser interesantes.

Se incluye un glosario con la definición de los términos y acrónimos más relevantes utilizados en la Memoria.

Se incluye la bibliografía con las referencias utilizadas y las URLs de las web relacionadas con el trabajo.

Se incluyen en anexos o en referencias a repositorios externos el contenido de los apartados extensos que constituyen el detalle del trabajo realizado.

2. Estado del arte

2.1. Visión retrospectiva sobre afinidad y algoritmos de predicción [1,2]

Las interacciones proteína-proteína se encuentran entre los procesos más importantes de la biología, desempeñando papeles fundamentales en el sistema inmunitario, las vías de señalización y la inhibición de enzimas. Los estudios sobre el proteoma han revelado que la mayoría de las proteínas interactúan con otras proteínas para llevar a cabo sus funciones celulares.

La caracterización experimental de la estructura de un complejo proteína-proteína es, sin embargo, difícil y no siempre exitosa. Para complementar los enfoques experimentales, a lo largo de los años se han desarrollado técnicas computacionales para la predicción de complejos proteicos, estimuladas por el experimento CAPRI (Critical Assessment of PRedicted Interactions). Entre los métodos computacionales de modelización de estructuras de complejos proteína-proteína se encuentran los métodos de acoplamiento ab initio, los métodos basados en la homología a partir de estructuras experimentales de complejos similares, y más recientemente, los métodos integradores basados en la información disponible, incluyendo los que usan técnicas de inteligencia artificial…

Una descripción computacional más completa de la interacción proteína-proteína requiere también algoritmos que puedan predecir las afinidades de unión. Aunque las funciones de energía para la predicción de la afinidad y la clasificación de las posturas de acoplamiento están relacionadas, a menudo se desarrollan específicamente para sus respectivos fines y hasta ahora han mostrado un rendimiento variable y bastante limitado. Ejemplos de áreas en las que se pueden mejorar las funciones de puntuación son las contribuciones entrópicas, los efectos del disolvente y la combinación óptima de términos. Para el desarrollo de algoritmos computacionales es esencial disponer de conjuntos de entrenamiento y prueba fiables y suficientemente amplios. La búsqueda de estructuras de complejos proteína-proteína en el Banco de Datos de Proteínas (PDB) es una ardua tarea desde el punto de vista computacional; las condiciones experimentales y la precisión de estas estructuras varían mucho y no siempre son fáciles de evaluar, como tampoco lo es la definición de la unidad biológica.

En relación con lo anterior, se han descrito varios trabajos de referencia que intentan recopilar un conjunto de datos fiables y bien comprendidos. En uno de los primeros esfuerzos para organizar de forma sistemática este tipo de conjunto de datos [1], se recopiló un conjunto no redundante de 144 complejos proteína-proteína que disponían de estructuras de alta resolución tanto para los complejos como para sus componentes no unidos, y cuyas constantes de disociación se habían medido por métodos biofísicos. . A este primer conjunto de datos se le denominó “affinity benchmark v1”. Este conjunto muestra gran diversidad en cuanto a las funciones biológicas que representa, incluyendo complejos enzima/inhibidor y enzima/sustrato, así como antígeno/anticuerpo y complejos en los que intervienen proteínas G y dominios extracelulares de receptores. Posteriormente, se describió una versión actualizada de complejos proteína-proteína con datos de afinidad o energía de unión, al que se denominó “affinity benchmark v2” [2]. Al igual que el primer conjunto de datos, los casos de referencia de docking y de afinidad consisten en estructuras no redundantes y de alta calidad de complejos proteína-proteína junto con las estructuras no unidas de sus componentes. En dicha actualización se añadieron 55 nuevos complejos a los casos de referencia de docking, 35 de los cuales tienen afinidades de unión medidas experimentalmente. Con ello, el conjunto de casos de referencia de afinidad actualizados contiene ahora 179 entradas.

Este conjunto se ha usado como punto de partida para desarrollar metodologías de predicción de afinidades de unión experimentales.

Desde los primeros tiempos de los estudios de interacción proteína-proteína, relacionar la estructura con la afinidad ha sido motivo de preocupación tanto para cristalógrafos como para bioquímicos y biofísicos. Sin embargo, estos estudios trataban sobre todo de sistemas individuales, y el primer intento de asociar las afinidades de unión con un conjunto de estructuras se debe a Horton y Lewis [10], que recogieron de la literatura 15 valores de ΔG de unión, y demostraron que podían ajustarse sumando las contribuciones de los grupos polares y no polares de la interfase. El ajuste tenía sólo tres coeficientes ajustables, y era notablemente bueno, produciendo un coeficiente de regresión lineal r = 0,96, y una diferencia absoluta media de 0,8 kcal mol^{-1} entre los valores ΔG calculados y observados. Sin embargo, la extensión de los resultados a otros complejos no produce buenos resultados. En informes posteriores se han utilizado conjuntos de datos más diversos, junto con modelos más elaborados y parámetros más ajustables. Ninguno ha logrado un ajuste tan bueno a los datos como Horton y Lewis, y podemos ver al menos dos razones para ello. La primera es la mala calidad de los conjuntos de datos, que contienen muchos valores de K_d incorrectos o asociados a entradas PDB erróneas, y otros que no se pueden rastrear hasta una medición real. La segunda razón es básica: los modelos se basan únicamente en características estructurales del complejo. Así, representan la reacción de asociación por su producto, e ignoran los reactantes o los cambios que pueda sufrir su estructura.

Usando los casos del “affinity benchmark v2”, se estudió la capacidad de una serie de algoritmos de predicción de docking previamente desarrollados, para predecir los valores de afinidad experimentales. Los valores de afinidad predichos mostraron buena correlación con las energías de unión experimentales (r = 0,52 en general y r = 0,72 para los complejos rígidos) [2]. Sin embargo, los rendimientos de estos algoritmos siguen sin dar resultados óptimos. En ellos, se exploran distintas mediciones, teniendo en cuenta los distintos tipos de complejos, como variables para establecer la energía de unión:

- el cambio en el área superficial enterrada, ΔASA, no se correlaciona bien con la energía de unión (r = -0,16);
- la energía de unión no se correlaciona altamente con I-RMSD (r = -0,24), y sólo se encuentra una pequeña mejora utilizando un modelo lineal mínimo que combina ΔASA e I-RMSD (r = 0,31)
- otra serie de métodos de predicción que incluyen la geometría y composición específicas de la interacción obtuvieron correlaciones globales de hasta r = 0,53, con un poder predictivo mucho mayor para los complejos rígidos, hasta r = 0,75, que en el caso de los casos flexibles llegó hasta r = 0,53.

Los métodos con mejores resultados se entrenaron utilizando el "benchmark affinity v1", aunque estas funciones produjeron correlaciones más bajas en los nuevos casos de referencia que la mejor correlación de r = 0,63 comunicada anteriormente para la referencia de afinidad original. Las correlaciones fueron menores para los potenciales estadísticos y las puntuaciones de acoplamiento.

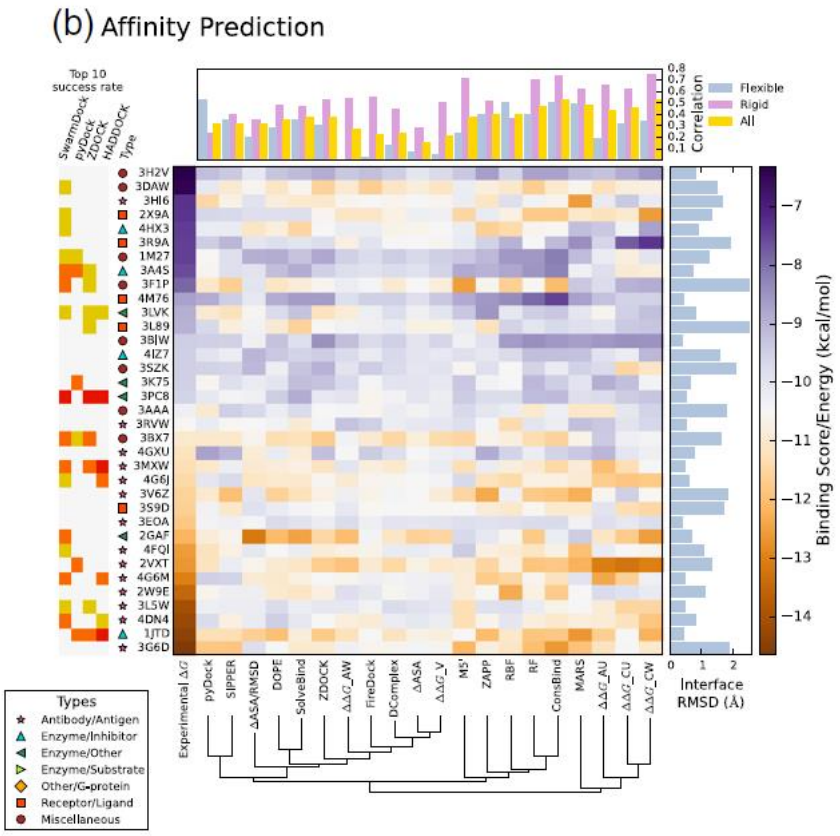

Figura 6: Evaluación de los métodos

Para algunos de los complejos, las predicciones fueron sistemáticamente pobres en todos los métodos. En la figura 6 se resume el comportamiento de algunos complejos, subestimando afinidades en algunos casos y sobreestimando afinidades en otros casos.

Entre los algoritmos desarrollados para la predicción de afinidad, podemos citar:

- ZAPP predice las energías libres de unión proteína-proteína utilizando una combinación lineal de nueve términos energéticos y una constante. Sólo un término utiliza las estructuras no unidas además de las estructuras complejas, mientras que los otros ocho términos sólo requieren la estructura compleja
- ConsBind es un método de predicción de afinidad basado en aprendizaje automático en el que la afinidad predicha es un consenso de cuatro aprendices: splines de regresión adaptativa multivariante, regresión random forest, interpolación de función de base radial y un árbol de regresión M5'. Los aprendices fueron entrenados usando 143 de las 144 afinidades en el anterior benchmark de afinidades con las 108 características extraídas de las estructuras ligadas usando el servidor web CCharPPI. No se utilizó la información de las estructuras no ligadas. La puntuación final consensuada es la media aritmética aritmética de los cuatro aprendices
- SolveBind es un método de predicción de afinidad de unión basado en el modelo de superficie global de Kastritis et al., que combina el número de átomos en la interfaz (NAtomsINT) y los porcentajes de residuos cargados y polares en la superficie de no-interacción (%AAchar NIS y %AApol NIS):
- Otros modelos como: modelo de afinidad mínima de Janin (ΔASA/RMSD), la superficie enterrada (ΔASA), los potenciales estadísticos DOPE y DComplex, las

puntuaciones de acoplamiento pyDock, SIPPER, ZDOCK y FireDock, así como los potenciales de contacto (ΔΔG_AW, ΔΔG_AU, ΔΔG_CW y ΔΔG_CU) y un modelo de energía superficial (ΔΔG_V) derivado de los datos de mutación

Como conclusión, se ve que en el rendimiento de un amplio conjunto de métodos punteros de predicción de afinidad de complejos proteína-proteína existentes las correlaciones con las afinidades obtenidas experimentalmente son inferiores a las registradas en versiones anteriores de la referencia. Por lo que, es interesante basándose en los puntos de referencia actualizados trabajar en la mejora de los algoritmos, que ayudarán a aumentar la comprensión de las interacciones biomoleculares.

2.2. Funciones de caracterización energética de complejos. Servidor CCharPPI [3]

Las estructuras atómicas de las interacciones proteína-proteína son fundamentales para comprender su papel en los sistemas biológicos, y se ha desarrollado una amplia variedad de funciones y potenciales biofísicos para su caracterización y la construcción de modelos predictivos. Estas herramientas están dispersas en multitud de programas independientes, y a menudo sólo están disponibles como parámetros de modelos que requieren una reimplementación. Esto supone un importante obstáculo para su adopción generalizada. CCharPPI integra muchas de estas herramientas en un único servidor web. Calcula hasta 108 parámetros, incluidos modelos de electrostática, desolvatación y enlace de hidrógeno, así como puntuaciones de empaquetamiento de interfaces y complementariedad, potenciales empíricos a varias resoluciones, potenciales de acoplamiento y funciones de puntuación compuestas.

CCharPPI es un servidor web (https://life.bsc.es/pid/ccharppi) que ha sido desarrollado por el grupo de investigación donde se ha llevado a cabo este trabajo [3]. El servidor permite calcular un gran número de descriptores y está disponible gratuitamente para uso académico no comercial.

Los descriptores se clasifican en las siguientes categorías:

- Potenciales de contacto/paso de residuos
- Potenciales dependientes de la distancia de los residuos
- Potenciales de contacto/paso atómicos
- Potenciales dependientes de la distancia atómica
- Términos constitutivos de potenciales estadísticos
- Funciones de puntuación compuestas
- Funciones de energía de solvatación
- Enlace de hidrógeno
- Van der Waals y electrostática
- Varios

Cada valor energético se ha implementado en base a la metodología descrita en su publicación original. Por ejemplo, para CP_BFKV del grupo de 'potenciales dependientes de la distancia de los residuos':

CP_BFKV. Contact potential, calculated between intermolecular residues. See Proteins 59(1):49 (2005) and BMC Bioinformatics 11:92 (2010) for details. Downloaded from http://gor.bb.iastate.edu/potential/

En conclusión, la reunión de muchos métodos diferentes para caracterizar las interacciones proteína-proteína puede acelerar la creación de prototipos de modelos

predictivos reproducibles, permitir a los usuarios mezclar y combinar diferentes formas funcionales para modelar fenómenos físicos, encontrar nuevos términos para sus funciones de puntuación y caracterizar sus complejos de interés.

2.3. Aprendizaje automático

¿Qué es el aprendizaje automático?

El aprendizaje automático (del inglés, machine learning) es una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. Frente a una programación tradicional, donde el programa determina el comportamiento de la ejecución, los modelos basados en algoritmos de aprendizaje automático "aprenden" mediante el uso de datos haciendo que su desempeño mejore con la experiencia; es decir, la habilidad no está explícita en el propio programa. De esta manera, el algoritmo procesa los datos, construye un modelo basado en esos datos y utiliza ese modelo como una hipótesis acerca del mundo orientado a resolver problemas en presencia de nuevos casos.

Tipos de sistemas basados en aprendizaje automático

Los sistemas basados en aprendizaje automático se pueden clasificar atendiendo a dos criterios: la tarea de aprendizaje o el tipo de algoritmo que usemos.

Desde el punto de vista de aprendizaje, tendríamos:

- Tareas de clasificación
- Tareas de predicción numérica
- Detección de patrones
- Detección de grupos (clusters)

Desde el punto de vista algorítmico, tendríamos:

- Algoritmos de aprendizaje supervisado
- Algoritmos de aprendizaje no supervisado
- Algoritmos de meta-aprendizaje

Se han propuesto [7] clasificaciones de algunos de los algoritmos existentes como los de la figura 7.

Model	Learning task
Supervised learning algorithms	
k-nearest neighbors	Classification
Naive Bayes	Classification
Decision trees	Classification
Classification rule learners	Classification
Linear regression	Numeric prediction
Regression trees	Numeric prediction
Model trees	Numeric prediction
Neural networks	Dual use
Support vector machines	Dual use
Unsupervised learning algorithms	
Association rules	Pattern detection
k-means clustering	Clustering
Meta-learning algorithms	
Bagging	Dual use
Boosting	Dual use
Random forests	Dual use

Figura 7: Algoritmos de ML (Lantz)

Puntos a tener en cuenta a la hora de su aplicación

Recientemente, se ha propuesto que, dado que los elementos esenciales son la selección de un modelo y el entrenamiento basado en un conjunto de datos, los puntos críticos a tener en cuenta son la posibilidad de escoger un mal modelo para el problema que se trata de resolver y que el conjunto de datos sea un mal conjunto de datos [8].

Desde el punto de vista de los datos, cuanto mayor sea el número de casos de los que dispongamos y cuanto más representativos sean, mejor calidad obtendremos del modelo. Asimismo, no todos los modelos son aplicables a cualquier problema, por lo que se deberán explorar los modelos que mejor aplicación tengan. Tendremos que vigilar:

- Calidad del dato
- Relevancia de las características medidas
- Ajuste excesivo o insuficiente del modelo

Entrenamiento y validación

La única forma de saber hasta qué punto un modelo se generalizará a nuevos casos es probarlo realmente en casos nuevos. Para contrastar los resultados una opción es dividir los datos en dos conjuntos: el conjunto de entrenamiento y el conjunto de prueba [8]. Como su nombre indica, el modelo se entrena con el conjunto de entrenamiento y se prueba con el conjunto de prueba. La tasa de error en los nuevos casos se denomina error de generalización

(o error fuera de la muestra) y, al evaluar el modelo en el conjunto de prueba, se obtiene una estimación de este error. Este valor indica el rendimiento del modelo en casos que nunca ha visto antes.

Si el error de entrenamiento es bajo (es decir, el modelo comete pocos errores en el conjunto de entrenamiento) pero el error de generalización es alto, significa que el modelo se está ajustando en exceso a los datos de entrenamiento.

Para evaluar modelos y compararlos, los entrenaremos y utilizaremos un conjunto de prueba que nos permita comparar su grado de generalización.

Para elegir los valores de hiperparámetro de los modelos, podemos como opción entrenar varios modelos diferentes utilizando valores distintos para este hiperparámetro y compararlos.

Pasos para la creación de modelos basados en aprendizaje automático

La figura 8 describe una serie de pasos propuestos [8].

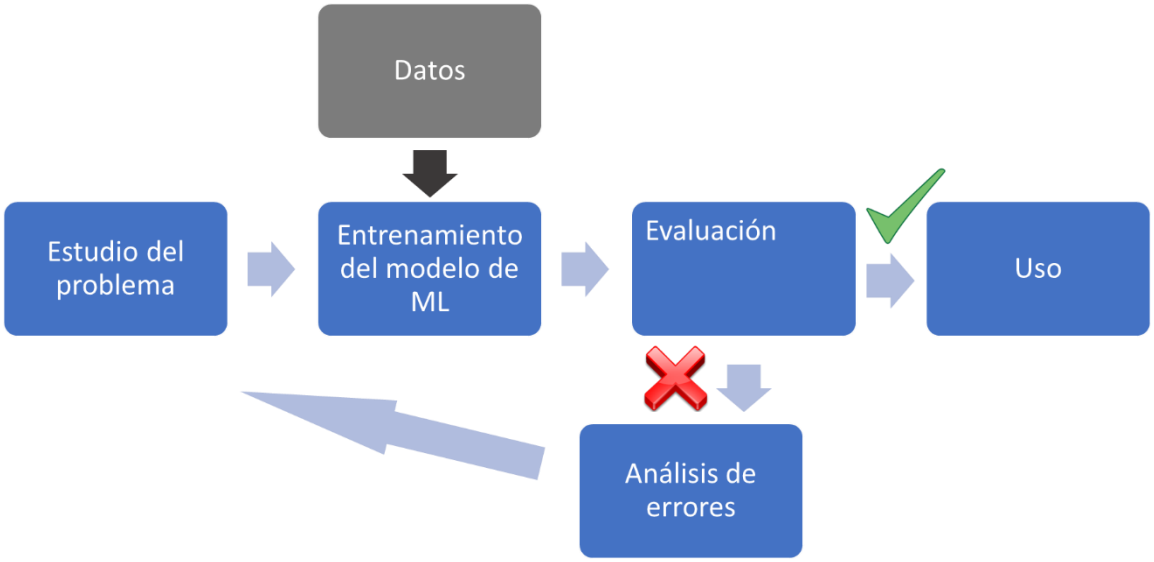

Figura 8: Pasos en ML (Géron)

Alternativamente, se ha descrito un proceso de aprendizaje de cinco pasos [7]:

1. Recogida de datos: El paso de recopilación de datos consiste en reunir el material de aprendizaje que un algoritmo utilizará para generar conocimiento procesable. En la mayoría de los casos, los datos deberán combinarse en una única fuente, como un archivo de texto, una hoja de cálculo o una base de datos

2. Exploración y preparación de datos: La calidad de cualquier proyecto de aprendizaje automático se basa en gran medida en la calidad de sus datos de entrada. Por lo tanto, es importante aprender más sobre los datos y sus matices durante una práctica denominada exploración de datos. Se requiere un trabajo adicional para preparar los datos para el proceso de aprendizaje. Esto implica arreglar o limpiar los llamados datos "desordenados", eliminar los datos innecesarios y recodificar los datos para que se ajusten a las entradas

3. Entrenamiento del modelo: En el momento en que los datos se han preparado para el análisis, es probable que tenga una idea de lo que es capaz de aprender de los datos. La tarea específica de aprendizaje automático elegida informará la selección de un algoritmo apropiado, y el algoritmo representará los datos en forma de modelo

4. Evaluación del modelo: Cada modelo de aprendizaje automático da lugar a una solución sesgada del problema de aprendizaje, lo que significa que es importante evaluar lo bien

que el algoritmo aprendió de su experiencia. Dependiendo del tipo de modelo utilizado, es posible que pueda evaluar la precisión del modelo utilizando un conjunto de datos de prueba, o puede que tenga que desarrollar medidas de rendimiento específicas para la aplicación prevista

5. Mejora del modelo: Si se necesita un mejor rendimiento, es necesario utilizar estrategias más avanzadas para aumentar el rendimiento del modelo. A veces puede ser necesario cambiar a otro tipo de modelo. Puede que sea necesario complementar los datos con datos adicionales o realizar un trabajo preparatorio adicional, como en el paso dos de este proceso. Una vez completados estos pasos, si el modelo parece funcionar bien, puede utilizarse para la tarea prevista. Según el caso, puede utilizar su modelo para proporcionar datos de puntuación para predicciones (posiblemente en tiempo real); para proyecciones de datos financieros; para generar información útil para el marketing o la investigación; o para automatizar tareas, como la entrega de correo o el vuelo de aviones. Los éxitos y fracasos del modelo desplegado podrían incluso proporcionar datos adicionales para entrenar a su aprendiz de próxima generación

2.4. Metodologías de última generación para el modelado estructural y energético de complejos entre proteínas

Pese a todo el trabajo hecho, los resultados obtenidos necesitan ser mejorados, por lo que nuevos métodos deben ser explorados.

Se plantea un enfoque de integración que complemente las actuales metodologías de modelado estructural basadas en IA que están presentando resultados que parecen muy prometedores, como lo demuestran los resultados de los experimentos de evaluación de metodologías CAPRI y CASP.

El Trabajo Fin de Máster (TFM) explora cómo mejorar algunos aspectos en los que estas metodologías se quedan cortas.

Dentro de estos aspectos, en el presente TFM se aplican funciones de puntuación de pyDock, ConsSurf y las funciones de CCharPPI, para mejorar la predicción de afinidad de complejos. Se aplican técnicas de optimización y aprendizaje automático, para encontrar las mejores combinaciones entre las funciones mencionadas.

Desde una perspectiva técnica, el planteamiento ha seguido estos pasos:

1) Datos de partida. Se parte del conjunto de estructuras "protein-protein docking benchmark 5.5" (https://zlab.umassmed.edu/benchmark/) que incluye datos de afinidad (Vreven et al 2015 J Mol Biol 427, 3031-3041).
2) Cálculos. Se calcularon diversas funciones de puntuación en las estructuras de los complejos de dicho benchmark, incluyendo pyDock (considerando sus términos individuales y la energía total), ConsSurf (https://consurf.tau.ac.il/consurf_index.php) y las funciones de CCharPPI (https://life.bsc.es/pid/ccharppi) (Moal et al. 2015 Bioinformatics 31, 123-125).
3) Análisis de correlaciones. Los valores obtenidos se compararon con los datos de afinidad experimentales y se llevó a cabo un análisis de la capacidad de predicción de los parámetros estudiados.
4) Desarrollo del modelo. Se buscaron combinaciones de variables individuales y modelos predictivos en base a métodos de aprendizaje automático, tipo random forest, árboles de decisión, etc.
5) Futuros desarrollos. Se consideró la aplicabilidad de las funciones y modelos obtenidos en modelos de docking o de AF-Multimer, usando tanto modelos individuales como ensamblados conformacionales.

El TFM se ha desarrollado en el entorno del Instituto de Ciencias de la Vid y del Vino (ICVV), aprovechando el conocimiento e infraestructura, aprovechando el conocimiento del grupo experto al que pertenece el tutor.

Todo ello ha permitido, a lo largo del desarrollo del TFM y de la obtención de resultados, la adquisición de competencias como:

- Inmersión en un entorno de trabajo Linux, métodos de visualización y análisis de estructuras de proteínas; manejo de bases de datos de estructuras (PDB); e interacciones (Interactome3D, STRING).
- Modelado de estructura de proteínas mediante metodologías basadas en inteligencia artificial y aprendizaje profundo (Deep Learning), como AlphaFold2. Estas metodologías han supuesto una auténtica revolución en el campo de la Bioinformática Estructural, ya que permiten modelar prácticamente todas las proteínas con estructura definida.
- Modelado estructural y energético de complejos entre proteínas. El grupo al que pertenece el tutor tiene una extensa trayectoria de desarrollo de herramientas computacionales para docking entre proteínas, como pyDock (Cheng et al 2007 Proteins 68:503-15; Rosell et al 2020 COSB 64, 59-65). En paralelo, los métodos basados en inteligencia artificial, como AF-MultiMer, que proporcionan predicciones de una gran calidad, en la mayor parte de complejos binarios entre proteínas.
- Se aplicaron funciones de puntuación de pyDock, ConsSurf y las funciones de CCharPPI, para mejorar la predicción de afinidad de complejos. Se aplicaron técnicas de optimización y aprendizaje automático, para encontrar las mejores combinaciones entre las funciones mencionadas.

3. Materiales y métodos

3.1. Entornos de trabajo

Como entornos de trabajo se ha utilizado SQL Server y Python bajo sistema operativo Windows. Python se utiliza bajo dos entornos: Visual Studio y Azure Data Studio.

En Python, se hacen uso de las siguientes librerías:

- pyodbc
- numpy
- pandas
- matplotlib
- scikit-learn

Como apoyo, se utilizan las herramientas ofimáticas de Microsoft Office 365: Excel, Word, Power Point,…

3.2. Datos de partida

Los datos de partida son los códigos PDB correspondientes a las estructuras 3D y los valores de afinidad de los complejos estudiados en los artículos sobre afinidad experimental. Existen dos versiones definidas, como se ha descrito en la sección 2.1 [1,2]. Cada una de las versiones aportan los siguientes complejos:

- 145 (144) complejos en el benchmark de afinidad v1 [1]; obtenidos desde fichero suministrado por el tutor
- 35 complejos en el benchmark de afinidad v2; procesados desde tabla del artículo original en PDF [2]

Además, obtenemos los ficheros PDB correspondientes a los complejos mediante los servicios de descarga de ficheros del 'RCSB Protein Data Bank (RCSB PDB)'.

3.3. Carga y limpieza de datos

Los complejos de la versión 1 se han obtenido desde el fichero de Excel 'Table_S1_V1.xls'. La incorporación de los complejos de la versión 2 se ha obtenido mediante la función de importación de datos de fichero PDF de Microsoft Excel. Una vez recuperados se han limpiado de manera manual hasta obtener la tabla que finalmente se utiliza (fichero 'AffinityBenchmark_v2.xlsx').

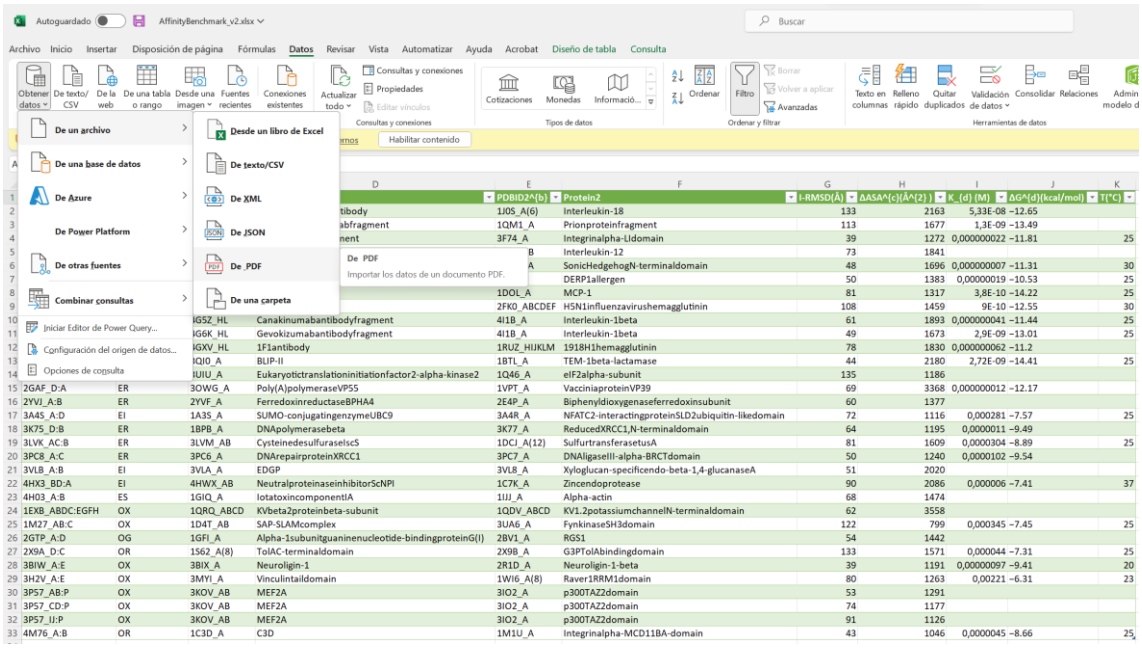

Figura 9: Importación de Excel desde PDF

3.4. Datos de trabajo en SQL Server

A partir de los datos en Excel, incorporamos los datos en tablas de SQL Server. Con estos datos, generamos scripts de cálculo de CCharPPI (ver capítulo Resultados). Los resultados del cálculo se incorporan, asimismo, para formar la base de complejos con los que trabajaremos.

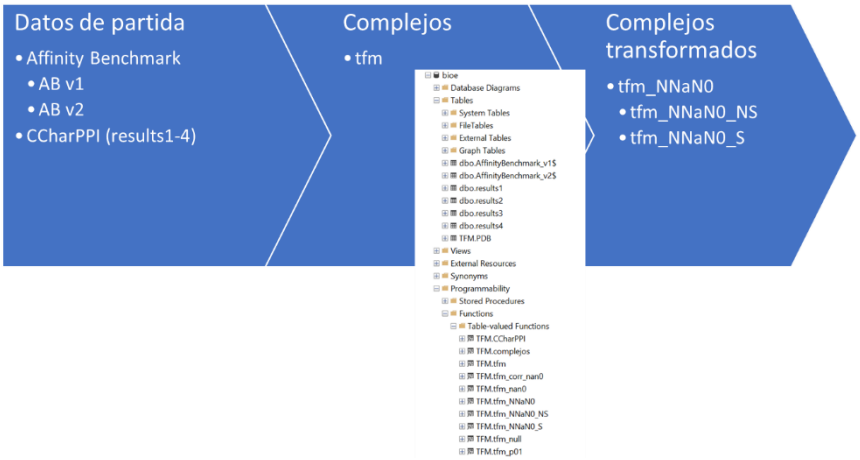

Figura 10: Estructuras SQL Server

4. Resultados

4.1. Cálculos CCharPPI

Partiendo de estos 180 complejos, procedemos a invocar el servicio por lotes de CCharPPI. Para ello formamos 3 lotes de 60 complejos cada uno. Obtenemos los ficheros PDB mediante los servicios de descarga de ficheros del 'RCSB Protein Data Bank (RCSB PDB)'.

Obtenemos valores calculados para 176 complejos. Repetimos el cálculo para los 4 sin valores, obteniendo valores para 2 de ellos. Los complejos sin cálculo son: '1UUG' y '3SGB'.

Tomamos, entonces, 178 complejos como base del estudio con valores calculados de potenciales energéticos mediante CCharPPI (hojas 'complejos', 'CCharPPI' y 'tfm' del fichero 'TFM.xlsx').

A continuación, estudiamos los descriptores de CCharPPI para detectar valores sin cálculo (nan). Con estas variables tenemos diversas opciones:

a) eliminar variable,
b) eliminar complejo,
c) sustituir por valor numérico: 0, valor medio,…

Las variables que contienen algún valor nan y su porcentaje sobre el total de valores son ('descriptores (nan)' del fichero 'TFM.xlsx') se describen en la figura 11.

descriptor	esnumero	cuenta	%nan
CP_ZPAIR_CB	0	15	8,43%
CP_ZLOCAL_CB	0	15	8,43%
CP_ZS3DC_CB	0	15	8,43%
CP_Z3DC_CB	0	15	8,43%
CP_EPAIR_CB	0	15	8,43%
CP_ELOCAL_CB	0	15	8,43%
CP_ES3DC_CB	0	15	8,43%
CP_E3DC_CB	0	15	8,43%
CP_E3D_CB	0	15	8,43%
CP_ZPAIR_MIN	0	14	7,87%
CP_ZLOCAL_MIN	0	14	7,87%
CP_ZS3DC_MIN	0	14	7,87%
CP_Z3DC_MIN	0	14	7,87%
CP_EPAIR_MIN	0	14	7,87%
CP_ELOCAL_MIN	0	14	7,87%
CP_ES3DC_MIN	0	14	7,87%
CP_E3DC_MIN	0	14	7,87%
CP_E3D_MIN	0	14	7,87%
AP_GOAP_ALL	0	5	2,81%
AP_GOAP_DF	0	5	2,81%
AP_GOAP_G	0	5	2,81%
NIPacking	0	1	0,56%
NSC	0	1	0,56%
ELE	0	1	0,56%
DESOLV	0	1	0,56%
VDW	0	1	0,56%
PYDOCK_TOT	0	1	0,56%
ODA	0	1	0,56%
PROPNSTS	0	1	0,56%
SIPPER	0	1	0,56%

Figura 11: Estudio de valores nan

Optamos por eliminar del estudio los descriptores que tienen más de 1 valor nan. Para los descriptores con una sola ocurrencia, los sustituimos por valores numéricos 0. Esto nos deja 87 descriptores del total de descriptores que calcula CCharPPI.

Por tanto, trabajaremos con los 178 complejos y los 87 valores calculados más la sustitución de los nan ('tfm_NNaN0' del fichero 'TFM.xlsx').

4.2. Análisis de correlaciones

Antes de explorar el desarrollo de modelos predictivos multi-paramétricos integrando los diferentes predictores de CCharPPI, es conveniente estudiar la capacidad predictiva de cada descriptor individual, analizando la correlación con los datos experimentales, posibles dependencias entre descriptores, o la capacidad clasificatoria de cada descriptor.

Así, en primer lugar, se ha analizado si existe correlación entre los valores obtenidos para el conjunto completo de casos por cada descriptor individual y los valores de afinidad experimentales (ΔG). La figura 12 muestra los valores de correlación (coeficiente de Pearson r) para cada descriptor individual. El descriptor con mejor correlación es CP_DDG_W con un

coeficiente (en valor absoluto) de 0,383578, que no es un valor muy alto. Los valores obtenidos para el coeficiente de correlación de Pearson se han incluido en el fichero 'TFM.xlsx' (columna 'correlación con ΔG').

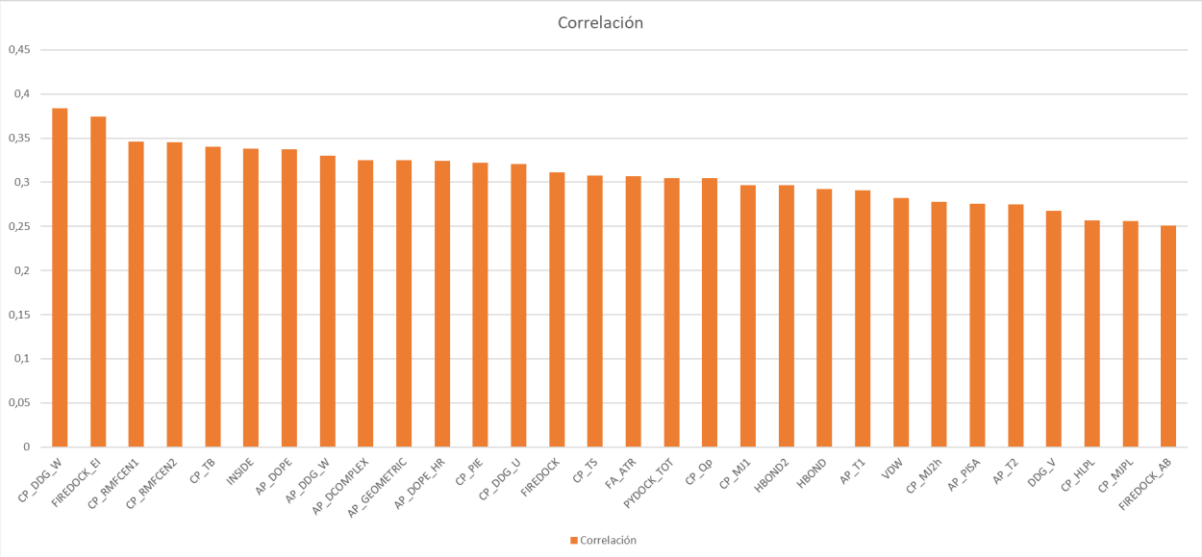

Figura 12: Correlación con ΔG

Por otro lado, para analizar si los descriptores son independientes o no, se ha calculado la correlación entre los valores obtenidos para el conjunto completo de casos por cada par de descriptores individuales. Aquí no estamos calculando correlaciones con los valores experimentales, sino entre los valores predichos por los descriptores. La figura 13 muestra los coeficientes de correlación de Pearson obtenidos cuando se comparan los descriptores individuales dos a dos, obteniendo un amplio rango de valores, desde pares con baja correlación (valores absolutos inferiores a 0,01; marcados en naranja) a pares con alta correlación (valores absolutos superiores a 0,8; marcados en verde) (columna 'correlación 2 a 2' del fichero 'TFM.xlsx').

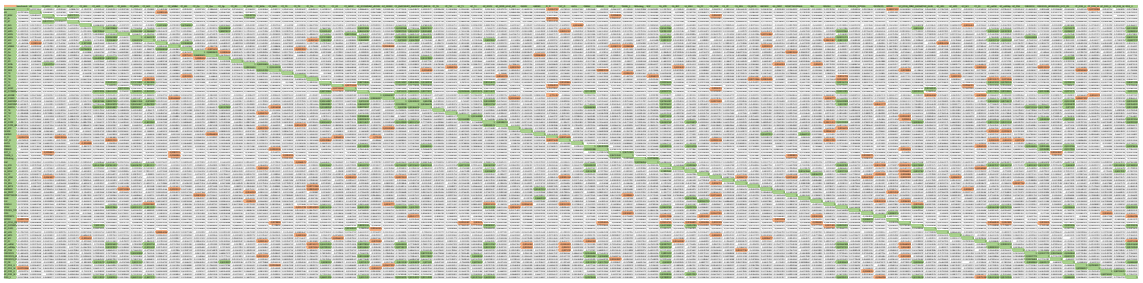

Figura 13: Correlación 2 a 2

Esta correlación se calcula mediante Python, con el método de dataframe de Pandas.

Esta correlación se aprecia, visualmente, mediante gráficos para casos de baja (figura 14) y alta correlación (figura 15):

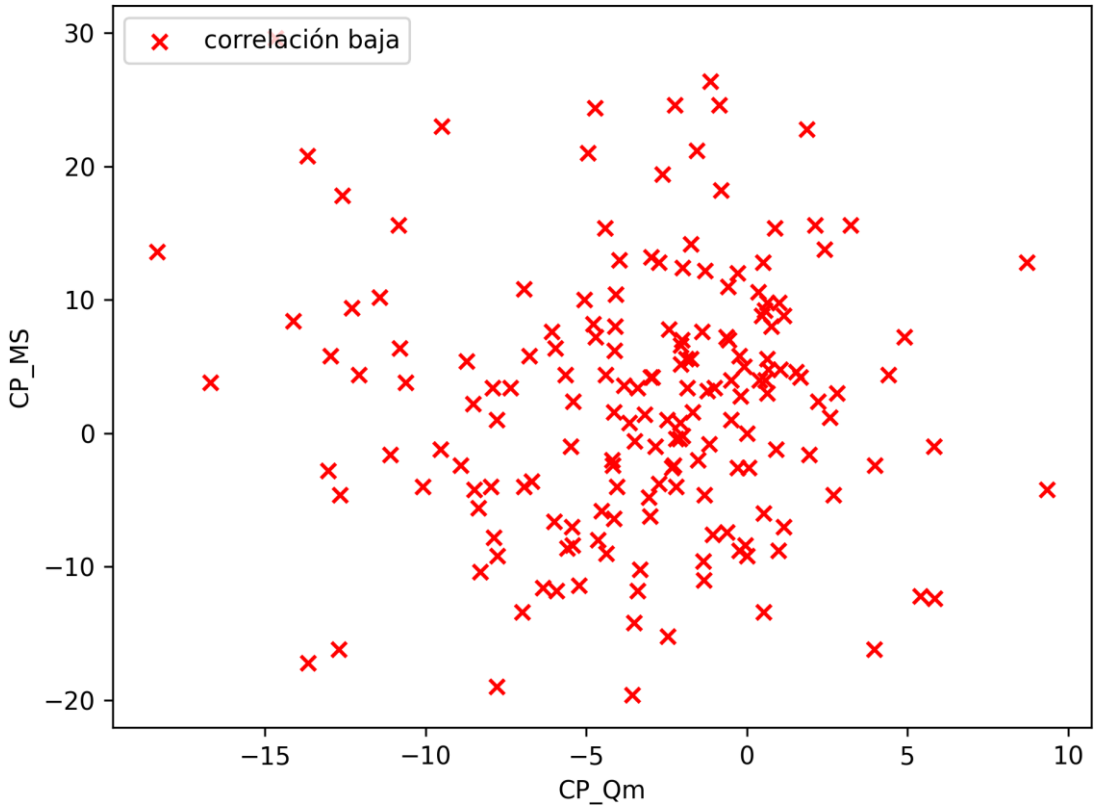

Figura 14: Ejemplo de correlación baja

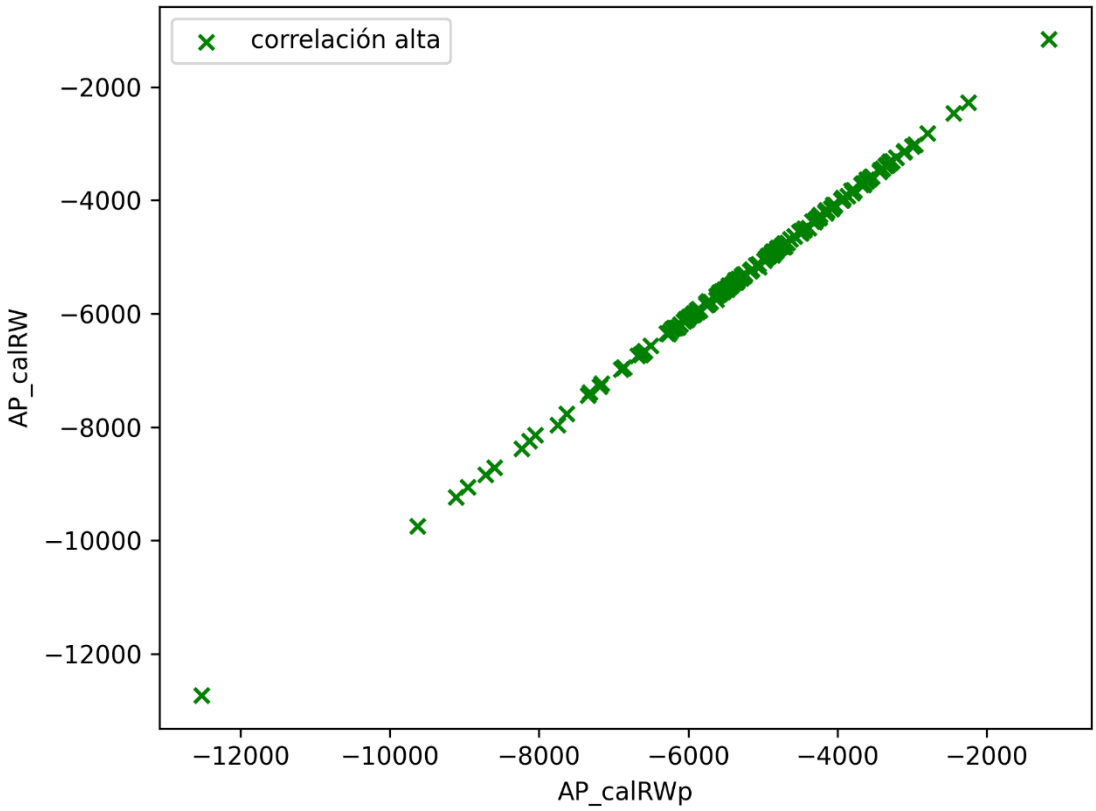

Figura 15: Ejemplo de correlación alta

4.3. Análisis de clasificación

Para análisis posteriores, vamos a clasificar las afinidades experimentales según su magnitud:

- Afinidad fuerte para valores negativos inferiores a -9
- Afinidad débil para valores negativos superiores (o iguales) a -9

En el análisis de clasificación, empezamos a estudiar la distribución de los valores calculados distinguiendo los casos según su clasificación de afinidad. Mediante la distribución de casos según su valor experimental vemos que los complejos clasificados no son linealmente separables.

En la figura 16 se puede ver las funciones de densidad y la representación mediante histogramas de los descriptores clasificados por grupo de afinidad.

Figura 16: Análisis de clasificación

En las figuras 17 y 18, vemos el caso del descriptor 'CP_BFKV'.

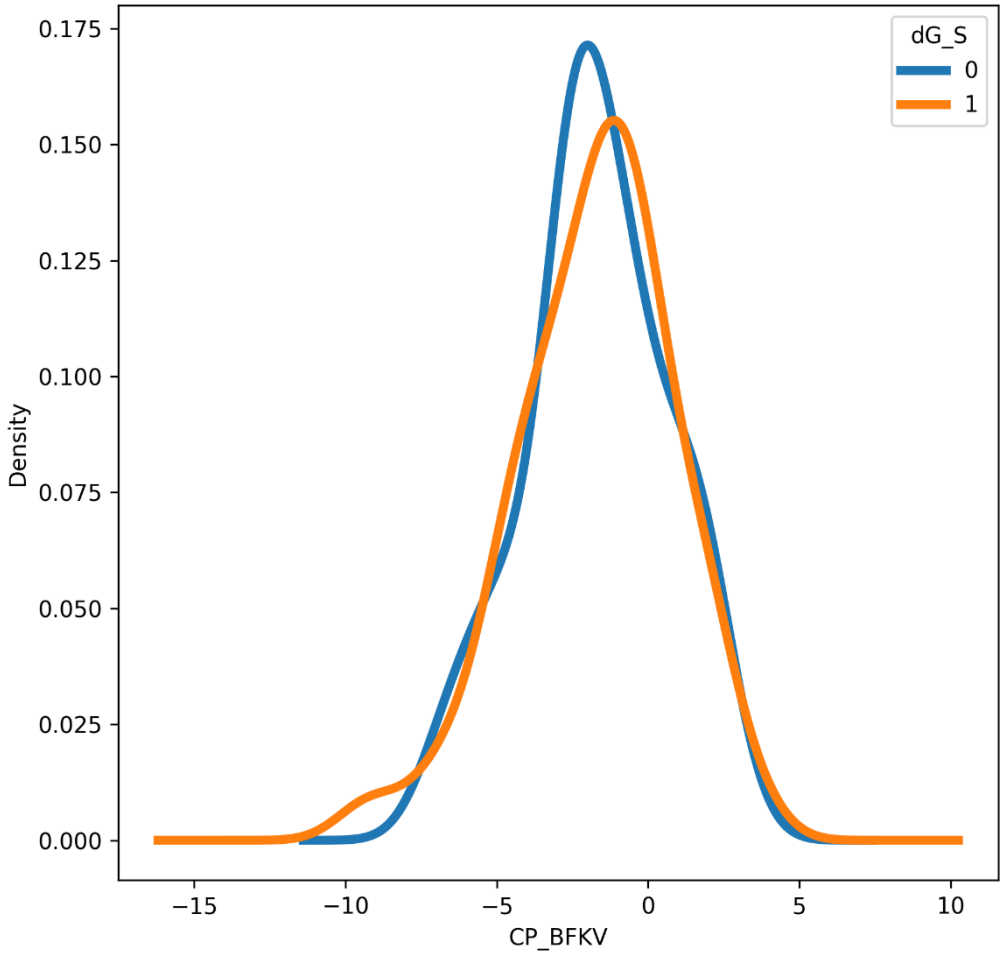

Figura 17: Función de densidad por grupos

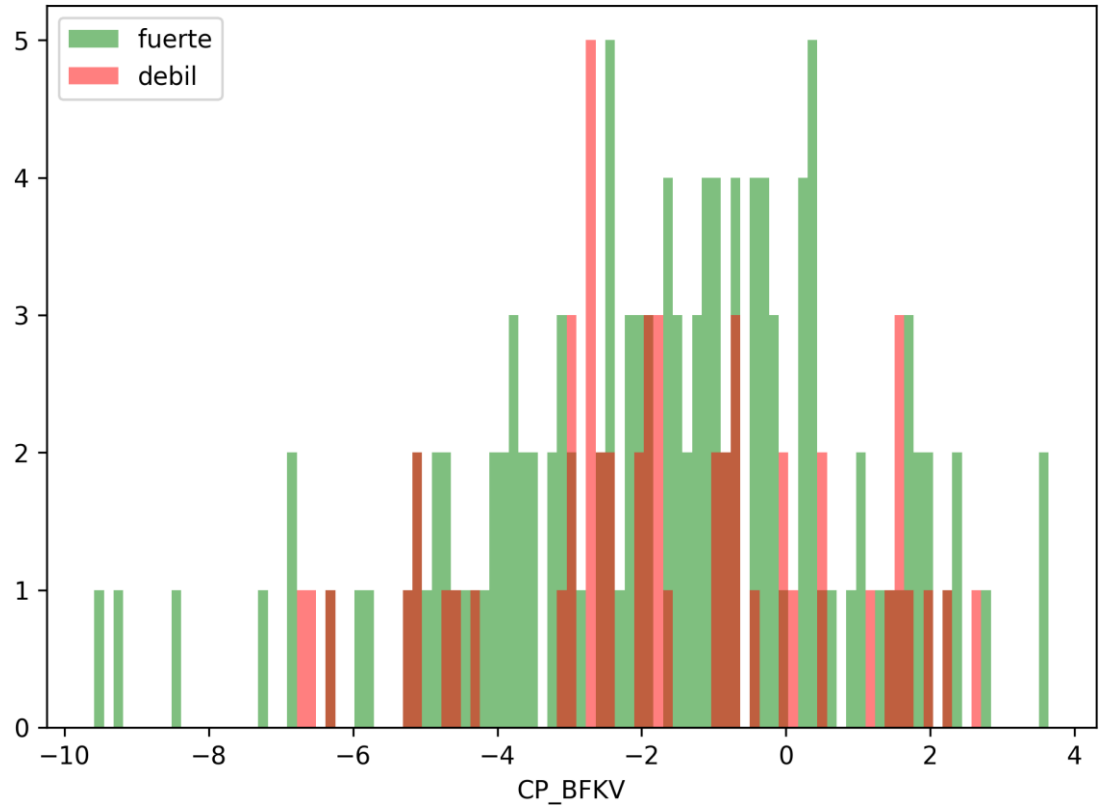

Figura 18: Histograma por grupos

Analizando pares de descriptores, vemos en las figura 19 y 20 que tampoco se aprecia separación en los grupos.

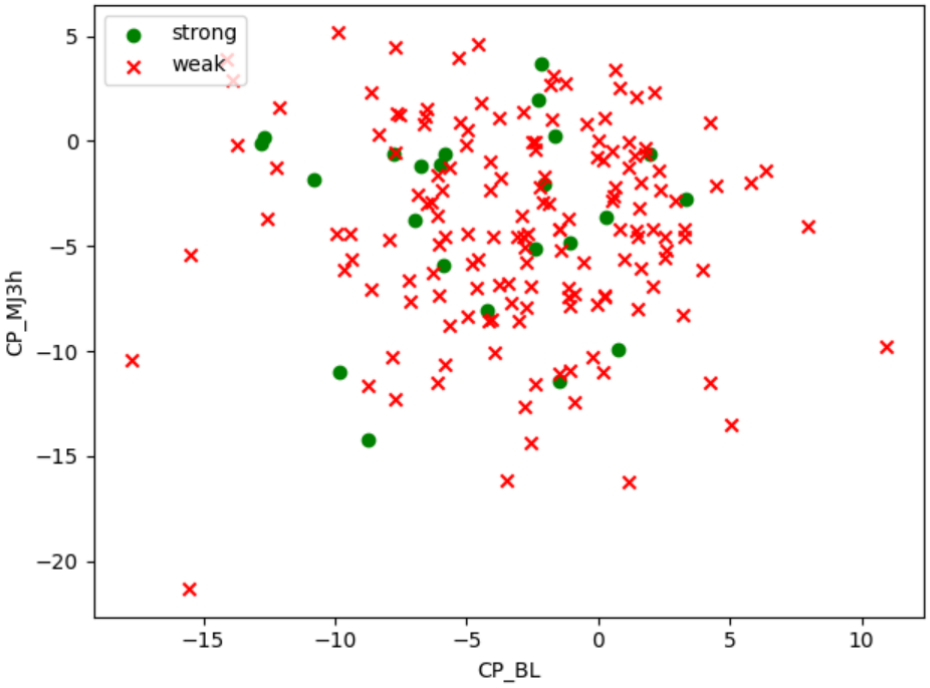

Figura 19: Distribución de afinidad según descriptores (1)

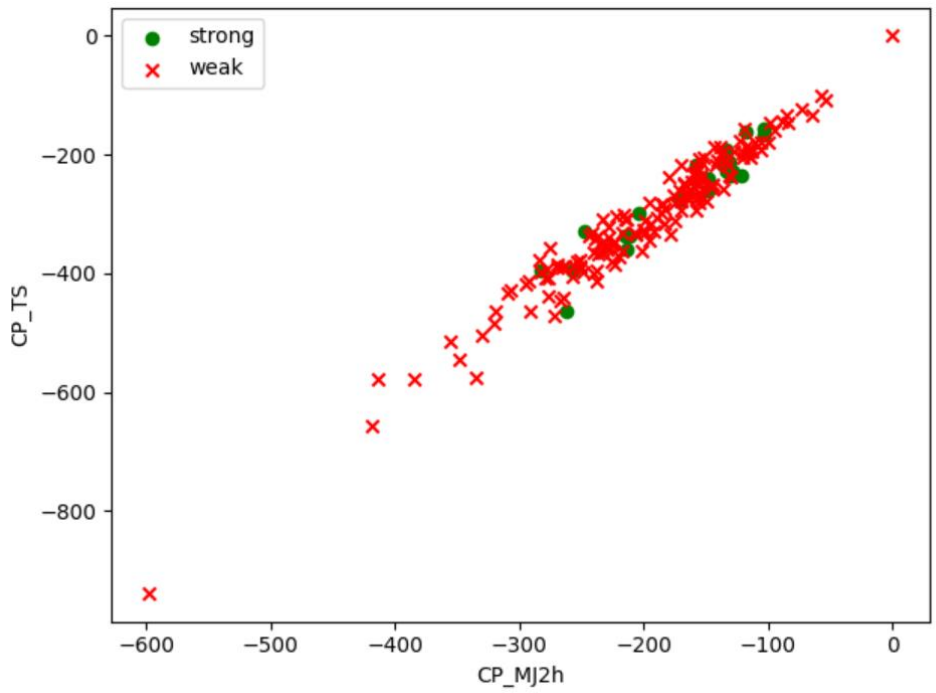

Figura 20: Distribución de afinidad según descriptores (2)

Por tanto, vemos necesario explorar modelos más sofisticados. Para ello, exploraremos métodos de aprendizaje automático.

4.4. Definición de nuevos métodos y técnicas a usar

De todos los posibles modelos de Machine Learning, el trabajo explora los identificados para poder dar un resultado según el objetivo planteado.

4.5. Desarrollo del modelo

En este punto, el trabajo explorará modelos de ML que nos pueden ayudar, tales como:

- análisis multivariables, reducción de variables (PCA), correlaciones multiparamétricas
- modelos de correlación
- modelos de clasificación sin supervisar: K-means
- modelos de clasificación supervisados: Random Forest, GBT

Modelos basados en K-means

K-Means es un algoritmo no supervisado de Clustering. Se utiliza cuando tenemos gran cantidad de datos sin etiquetar. El objetivo de este algoritmo es el de encontrar “K” grupos (clusters) entre los datos a partir de su análisis.

El algoritmo trabaja manera iterativa para asignar a cada punto, cuyas coordenadas se forman en base a las características, uno de los “K” grupos identificados por sus características. Son agrupados en base a la similitud de sus características. Como resultado de ejecutar el algoritmo tendremos:

- Los “centroids” de cada grupo que serán unas “coordenadas” de cada uno de los K conjuntos que se utilizarán para poder etiquetar nuevas muestras

- Etiquetas para el conjunto de datos de entrenamiento. Cada etiqueta perteneciente a uno de los K grupos formados

Vamos a intentar agrupar los complejos en base a sus características. Fijamos 3 descriptores de trabajo para poder visualizar y analizar resultados: 'CP_BFKV','CP_BL' y 'CP_BT'. En la figura 21 se cruzan los descriptores de los complejos para intentar observar su agrupación y la relación con sus afinidades.

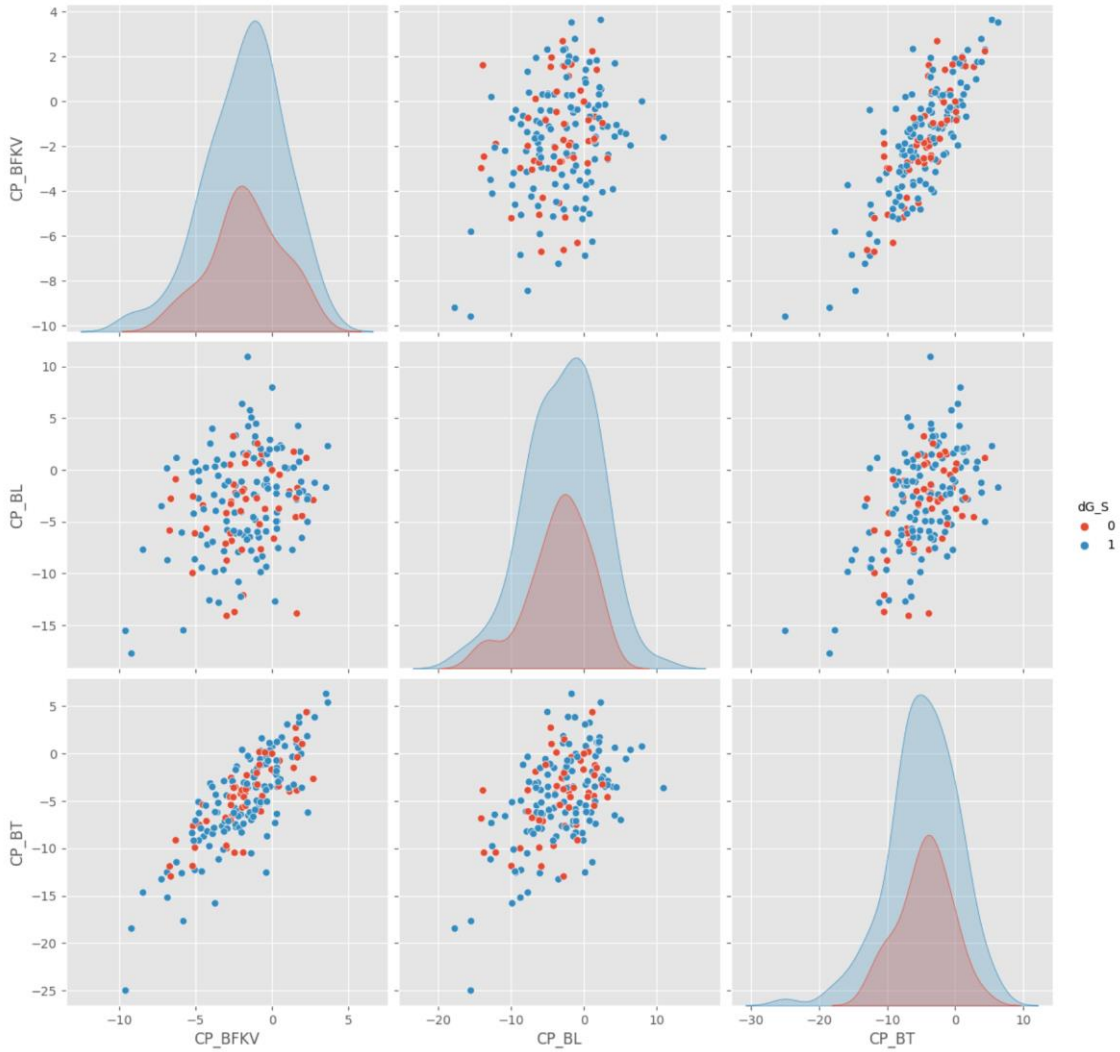

Figura 21: Cruce de descriptores

Los puntos de partida representados en la figura 22, serán los que el algoritmo intente agrupar por categorías identificadas por el análisis.

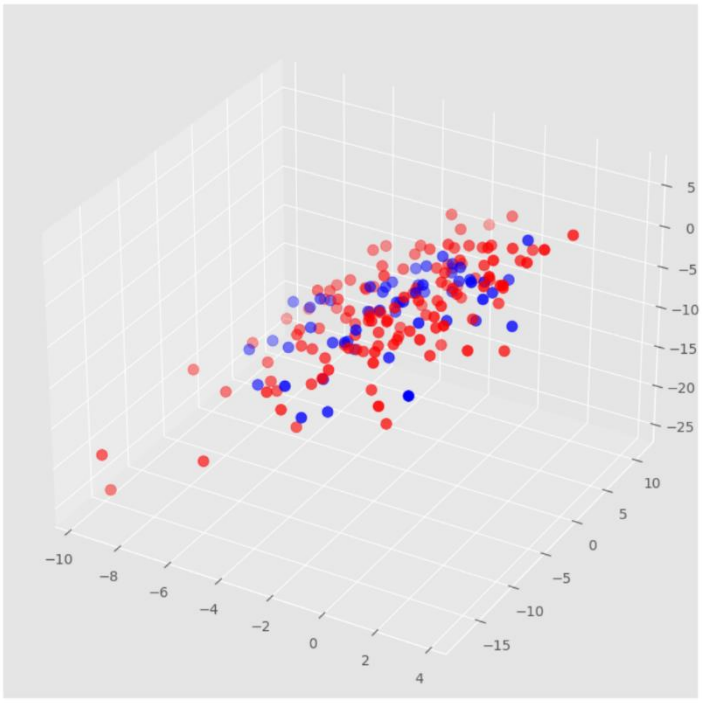

Figura 22: Puntos de partida

Para la obtención del valor K a aplicar, generamos una curva que nos indica un punto de inflexión. En este caso, definimos K=5 en base a la curva de la figura 23.

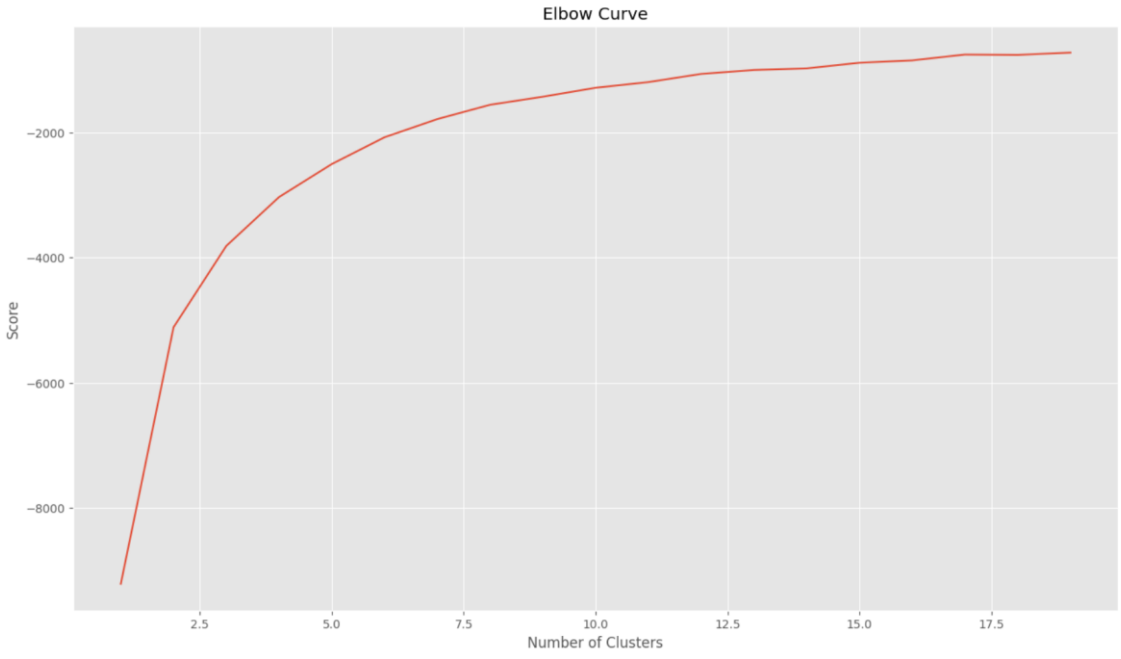

Figura 23: Curva para determinar K

Ejecutando el algoritmo K-means obtenemos 5 clusters. Estos clusters se representan en la figura 24. En la figura 25, el resultado se proyecta en 2 dimensiones para ayudar en la visualización de grupos.

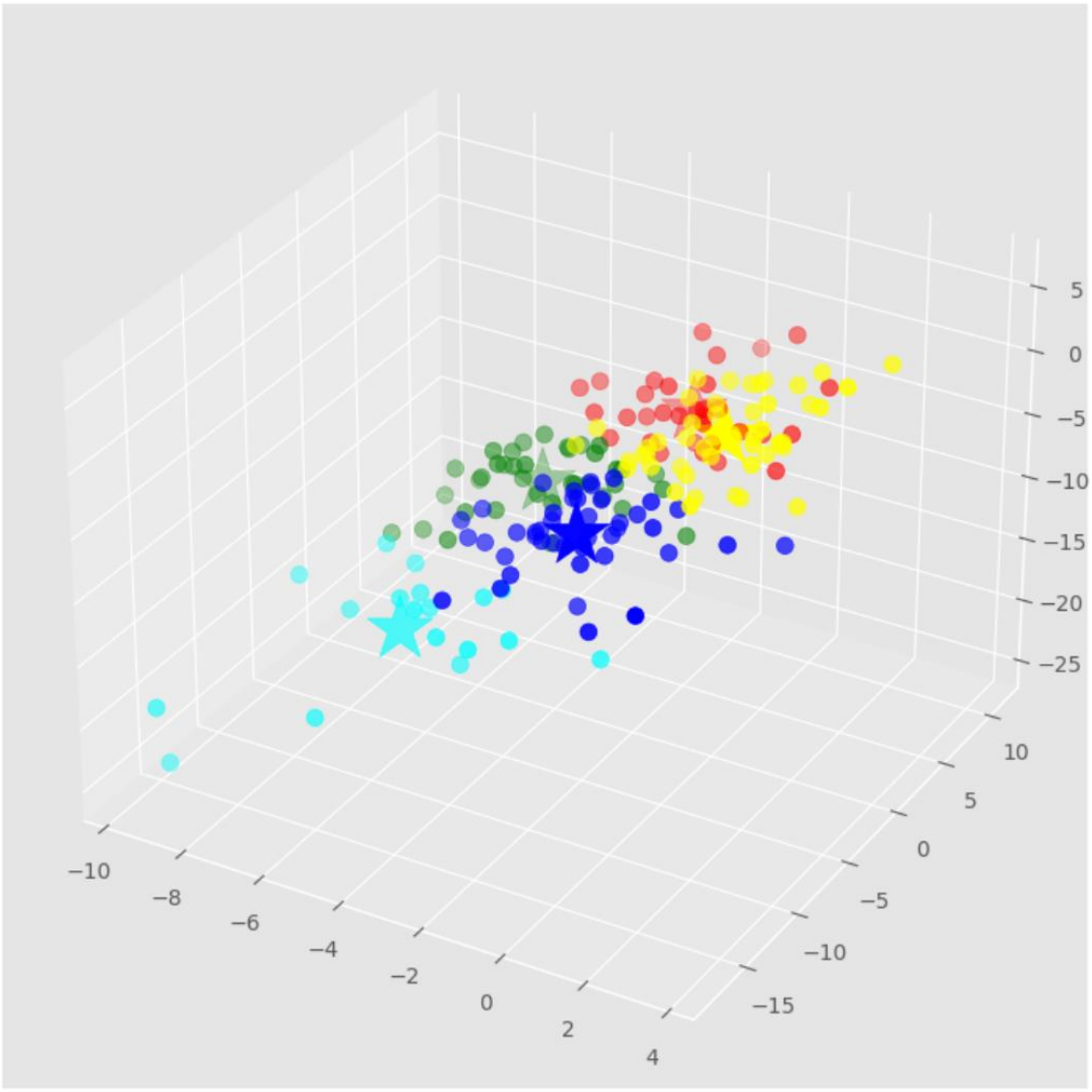

Figura 24: Agrupaciones K-means

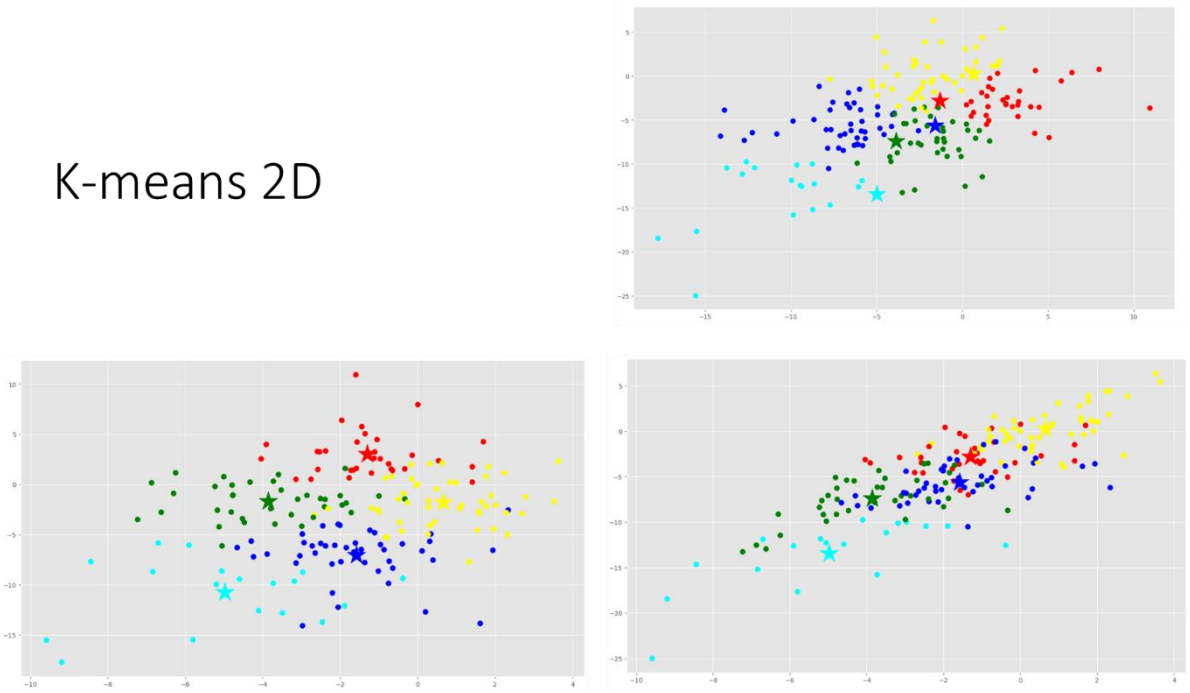

Figura 25: Proyecciones 2D

Este modelo permite predecir la agrupación de nuevos complejos. El análisis de agrupaciones permite detectar complejos que presentan características comunes, lo que puede conducir a un tratamiento diferenciado para la determinación de la afinidad del complejo.

Modelos basados en Random Forest

Random Forest es un tipo de ensamble en Machine Learning en donde se combinan diversos árboles de decisión y la salida de cada uno se cuenta como un voto. La respuesta del Radndom Forest es la opción más votada.

Random Forest, al igual que el árbol de decisión, es un modelo de aprendizaje supervisado para clasificación (aunque también puede usarse para problemas de regresión).

Los árboles de decisión son representaciones gráficas de posibles soluciones a una decisión. Son algoritmos de aprendizaje supervisado y pueden realizar tareas de clasificación o regresión.

PCA como herramienta para reducción de dimensiones

El PCA (Principal Component Analysis) es una herramienta que nos permite reducir dimensiones. Dentro de los descriptores de CCharPPI, es posible que algunos de ellos sean menos importantes y no aporten demasiado valor a la predicción. Cuando vimos la correlación 2 a 2, algunos de los descriptores estaban correlacionados. Además, el uso de un gran número de variables en los modelos podría favorecer el overfitting, lo que llevaría a modelos menos generalizables.

La reducción de dimensiones se puede hacer mediante:

- La eliminación por completo de dimensiones
- Extracción de características

En la eliminación debemos tener la certeza de que estamos quitando dimensiones poco importantes. En la extracción de características actuaremos mediante combinación de las existentes.

Realizando un análisis PCA de los descriptores, podemos ver en la figura 26 que un número menor de descriptores, del orden de 20, acumularía una variabilidad explicada cercana al 90%.

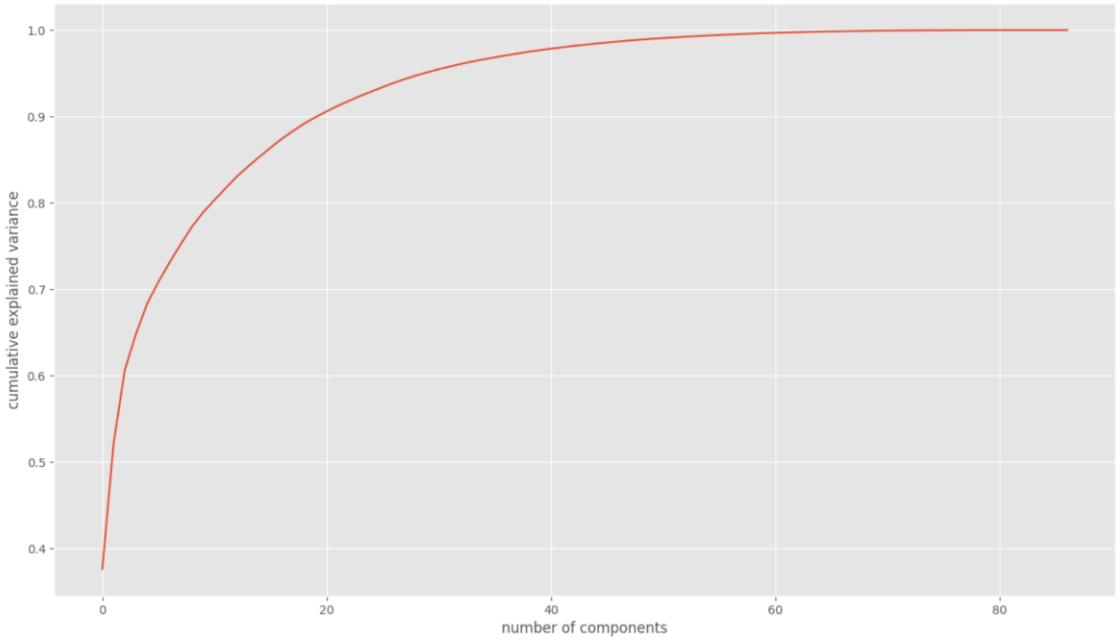

Figura 26: Variabilidad acumulada

Representando los dos componentes (descriptores) principales veríamos tener una idea de sus predicciones (figura 27).

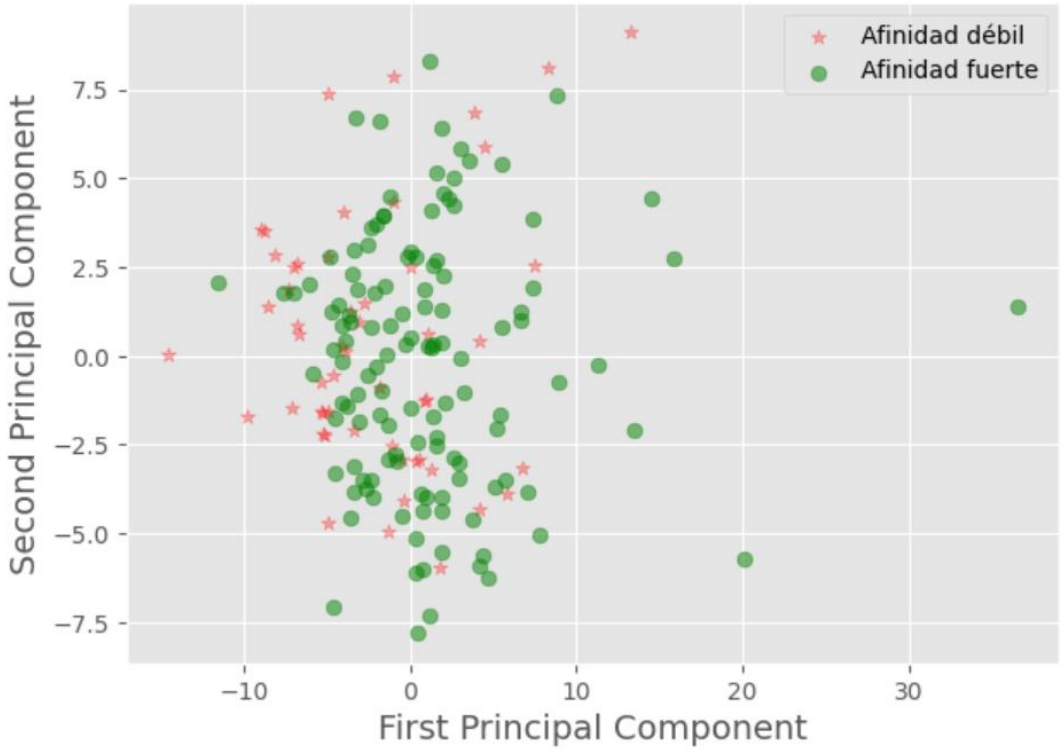

Figura 27: Visión reducida a dos componentes

5. Conclusiones y trabajos futuros

5.1. Conclusiones

El camino iniciado por el presente TFM supone una vía de gran perspectiva para explorar metodologías de última generación para el modelado estructural y energético de complejos entre proteínas. El uso de algoritmos de Machine Learning para la clasificación y predicción de la afinidad experimental a partir de los valores calculados de potencial de CCharPPI constituye un amplio campo en el que determinar el rendimiento de los posibles modelos a desarrollar.

El número elevado de descriptores calculados por CCharPPI hace que el descubrimiento de modelos de correlación con la afinidad no sea viable sin el uso de algoritmos de aprendizaje automático soportados por ordenador. Estos algoritmos permitirán el descubrimiento de patrones en los complejos y sus descriptores que de otra manera sería muy complicado determinar.

Debido al tiempo dedicado a la adquisición del conocimiento de contexto en biología estructural y a la comprensión de los aspectos del problema planteado, la exploración de modelos de aprendizaje automático no ha sido lo profunda que hubiera sido necesario para evaluar los modelos posibles.

5.2. Trabajos futuros

Se pueden explorar nuevos modelos, tomando como punto de partida el análisis hecho en el presente TFM, mediante el uso de los algoritmos identificados, trabajando sobre la hiperparametrización o bien mediante el uso de otros algoritmos. A medida que aparezcan nuevos complejos con afinidad experimental, estos podrán incorporarse como casos de entrenamiento o para validación de los modelos, haciendo que mejore la capacidad de aprendizaje de los algoritmos.

6. Glosario

- Complejo proteína-proteína: un complejo proteico o complejo multiproteico es un grupo de dos o más cadenas polipeptídicas asociadas
- Docking: en el campo del modelado molecular, el acoplamiento es un método que predice la orientación preferida de una molécula respecto a otra cuando un ligando y una diana se unen para formar un complejo estable
- Afinidad: la afinidad de unión es la fuerza de la interacción de unión entre una única biomolécula (por ejemplo, proteína o ADN) con su ligando/socio de unión (por ejemplo, fármaco o inhibidor)

7. Bibliografía

7.1. Documentos

1. Kastritis PL, Moal IH, Hwang H, Weng Z, Bates PA, Bonvin AMJJ, Janin J (2011). A structure-based benchmark for protein–protein binding affinity-Wiley. Protein Science V20: 482-491

2. Vreven T, Moal IH, Vangone A, Pierce BG, Kastritis PL, Torchala M, Chaleil R, Jiménez-García B, Bates PA, Fernández-Recio J et al. (2015). Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. J Mol Biol, 427:3031-3041

3. Moal IH, Jiménez-García B, Fernández-Recio J. (2015) CCharPPI web server: computational characterization of protein–protein interactions from structure. Bioinformatics, 31:123-125

4. Cheng,T.M. et al. (2007). pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. Proteins, 68, 503–515.

5. Fernández-Recio J, Rosell M. (2020). Docking approaches for modeling multi-molecular assemblies. Current Opinion in Structural Biology, 64:59–65

6. Kessel A, Ben-Tal N. (2018). Introduction to Proteins Structure, Function, and Motion, second edition, CRC

7. Lantz B. (2019). Machine Learning with R, third edition. Packt

8. Géron A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, third edition. O'Reilly Media, Inc.

9. Barbany M. (). Biología structural (apuntes asignatura del Máster en Bioinformática y Bioestadística). PID_00192773. UOC

10. Horton N, Lewis M (1992) Calculation of the free energy of association for protein complexes. Protein Sci 1:169–181.

URLs

U1. Protein-protein docking benchmark 5.5: https://zlab.umassmed.edu/benchmark/

U2. pyDock: https://life.bsc.es/pid/pydock/

U3. ConsSurf: https://consurf.tau.ac.il/consurf_index.php

U4. Funciones de CCharPPI: https://life.bsc.es/pid/ccharppi

U5. AlphaFold-Multimer: https://www.deepmind.com/publications/protein-complex-prediction-with-alphafold-multimer

U6. ICVV: https://www.icvv.es/

U7. Protein Data Bank: https://www.rcsb.org/

8. Anexos

8.1. Repositorio en GitHub

Se ha creado un repositorio en GitHub para depositar material de trabajo y resultados del trabajo. El repositorio se accede en: https://github.com/mjimenezcaruoc/MByB-TFM

8.2. Algunos ejemplos de código Python

Para el cálculo de correlación:

```
import numpy as np

from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt

%matplotlib inline

df_corr = df.corr(method='pearson')
print(df_corr.columns)
print(df_corr.iloc[0])
#print(df_corr.head(0))

print(df.corr(method='pearson')["dG"]["CP_BFKV"])

from pathlib import Path
filepath = Path('../tmp/df_corr.csv')
df_corr.to_csv(filepath)

```

Para la visualización de correlación entre descriptores:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

lista_descriptores = [["CP_Qm","CP_MS","correlación baja","red"],["AP_calRWp","AP_calRW","correlación alta","green"]]
for idesc in lista_descriptores:
    print(idesc)
    for iclasif in idesc:
        print(iclasif)

    descriptor1 = idesc[0]
    descriptor2 = idesc[1]

    y = df[[descriptor2]].values
    X = df[[descriptor1]].values

    # plot data
    plt.scatter(y, X, color=idesc[3], marker='x', label=idesc[2])

    plt.xlabel(descriptor1)
    plt.ylabel(descriptor2)
    plt.legend(loc='upper left')

    plt.tight_layout()
    plt.savefig('../tmp/corr_'+idesc[2]+'.png', dpi=300)
    plt.show()

```

Para el análisis de la clasificación:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

lista_descriptores = ["CP_BFKV","CP_BL"]
top = "10"
query = "select top(" + top + ") col.name from sys.objects as tab inner join sys.columns as col on tab.object_id = col.object_id" + \
    " where tab.name = 'tfm_NNaN0' and col.name not in ('Complex','benchmark','dG') order by column_id;"
#print(query)
ld = pd.read_sql_query(query, cnxn)
#print(ld.head())
col_one_list = ld['name'].tolist()
#print(f"\ncol_one_list:\n{col_one_list}\ntype:{type(col_one_list)}")
lista_descriptores = col_one_list
for idesc in lista_descriptores:
    #print(idesc)

    query = "SELECT [Complex],[dG_S],[" + idesc + "] FROM [TFM].[tfm_NNaN0_S] () ORDER BY [Complex];"
    #print(query)

    df = pd.read_sql_query(query, cnxn)
    #print(df.head(5))

    # converting data into wide-format
    data_wide = df.pivot(columns='dG_S', values=idesc)
    data_wide.head()

    # calling density() to make
    # multiple density plot
    data_wide.plot.density(figsize = (7, 7), linewidth = 4)

    plt.xlabel(idesc)
    plt.savefig('../tmp/clasif_'+idesc+'.png', dpi=300)
    plt.show()

```

Generación del modelo K-means:

```
K=5 # por interpretación del gráfico anterior

kmeans = KMeans(n_clusters=K).fit(X)
centroids = kmeans.cluster_centers_
print(centroids)
```

Para la predicción de clusters:

```
# Predicting the clusters
labels = kmeans.predict(X)
# Getting the cluster centers
C = kmeans.cluster_centers_
colores=['red','green','blue','cyan','yellow']
asignar=[]
for row in labels:
    asignar.append(colores[row])

fig = plt.figure()
#ax = Axes3D(fig)
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=asignar,s=60)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
```

Para el análisis de reducción de dimensiones mediante PCA:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

#cargamos los datos de entrada
query = "SELECT dG_S, CP_BFKV,CP_BL,CP_BT,CP_GKS,CP_HLPL,CP_MJPL,CP_MJ3h,CP_MJ2h,CP_MJ1,CP_MJ2,CP_MSBM,CP_MS,CP_Qa,CP_Qm,CP_Qp" + \
  ",CP_RO,CP_SKOb,CP_SKOa,CP_SJKG,CP_TD,CP_TEl,CP_TEs,CP_TS,CP_VD,CP_SKOIP,AP_DCOMPLEX,AP_dDFIRE,AP_DFIRE2,CP_RMFCEN1" + \
  ",CP_RMFCEN2,CP_RMFCA,CP_TB,CP_TSC,AP_T1,AP_T2,AP_DOPE,AP_DOPE_HR,AP_ACE,INSIDE,HBOND,PI_PI,CAT_PI,ALIPH,ZRANK,ZRANK2" + \
  ",ROT_S,TRANS_S,NIPacking,NSC,FA_ATR,FA_REP,LK_SOLV,FA_PP,CG_VDW,CG_PP,CG_ENV,CG_BETA,HBOND2,AA_PROP,ROSETTADOCK,NHB" + \
  ",ELE,DESOLV,VDW,PYDOCK_TOT,ODA,PROPNSTS,SIPPER,AP_OPUS_PSP,AP_GEOMETRIC,AP_DARS,AP_URS,AP_MPS,AP_W1,CP_D1,AP_calRW" + \
  ",AP_calRWp,AP_PISA,FIREDOCK,FIREDOCK_AB,FIREDOCK_EI,CP_PIE,CP_DDG_U,CP_DDG_W,AP_DDG_U,AP_DDG_W,DDG_V FROM [bioe].[TFM].[tfm_NNaN0_S] ();"

dataframe = pd.read_sql_query(query, cnxn)
print(dataframe.head(5))

#normalizamos los datos
scaler=StandardScaler()
df = dataframe.drop(columns=["dG_S"]) # quito la variable dependiente "Y"
scaler.fit(df) # calculo la media para poder hacer la transformacion
X_scaled=scaler.transform(df) # Ahora si, escalo los datos y los normalizo

#Instanciamos objeto PCA y aplicamos
pca=PCA(n_components=87) # Otra opción es instanciar pca sólo con dimensiones nuevas hasta obtener un mínimo "explicado" ej.: pca=PCA(.85)
pca.fit(X_scaled) # obtener los componentes principales
X_pca=pca.transform(X_scaled) # convertimos nuestros datos con las nuevas dimensiones de PCA

print("shape of X_pca", X_pca.shape)
expl = pca.explained_variance_ratio_
print(expl)
print('suma:',sum(expl[0:5]))
#Vemos que con 5 componentes tenemos algo mas del 85% de varianza explicada

#graficamos el acumulado de varianza explicada en las nuevas dimensiones
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')
plt.show()

#graficamos en 2 Dimensiones, tomando los 2 primeros componentes principales
Xax=X_pca[:,0]
Yax=X_pca[:,1]
labels=dataframe["dG_S"].values
cdict={0:'red',1:'green'}
labl={0:'Afinidad débil',1:'Afinidad fuerte'}
marker={0:'*',1:'o'}
alpha={0:.3, 1:.5}
fig,ax=plt.subplots(figsize=(7,5))
fig.patch.set_facecolor('white')
for l in np.unique(labels):
    ix=np.where(labels==l)
    ax.scatter(Xax[ix],Yax[ix],c=cdict[l],label=labl[l],s=40,marker=marker[l],alpha=alpha[l])

plt.xlabel("First Principal Component",fontsize=14)
plt.ylabel("Second Principal Component",fontsize=14)
plt.legend()
plt.show()

```