Predicción de la interacción de proteínas relacionadas con el Alzheimer a partir de su estructura primaria.

Alumno: Carlos Pérez López

Máster Universitario en Bioinformática y Bioestadística

Área del trabajo: Bioinformática Farmacéutica

Tutor: Melchor Sanchez Martinez
Profesor responsable de la asignatura: Javier Luis Cánovas Izquierdo

Fecha: 08/01/2020

A) Creative Commons:

B) GNU Free Documentation License (GNU FDL)

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Predicción de la interacción de proteínas relacionadas con el Alzheimer a partir de su estructura primaria.*
Nombre del autor:	*Carlos Pérez López*
Nombre del consultor/a:	*Melchor Sánchez Martínez*
Nombre del PRA:	*Javier Luis Cánovas Izquierdo*
Fecha de entrega (mm/aaaa):	01/2020
Titulación:	*Máster Universitario en Bioinformática y Bioestadística*
Área del Trabajo Final:	Bioinformática Farmacéutica
Idioma del trabajo:	*Castellano*
Palabras clave	*Interacción entre proteínas, Support Vector Machine, Random Forest*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La enfermedad del Alzheimer (EA) es una enfermedad neurodegenerativa que afecta a una gran cantidad de personas en la actualidad. Todavía se está trabajando en multitud de terapias, sin embargo, los estudios para la elaboración de nuevos medicamentos, son procesos muy costosos; por lo que se está recurriendo a técnicas computacionales de machine learning para abaratar costes. En este trabajo se van a entrenar modelos de machine learning para intentar predecir si dos proteínas interaccionan o no. Para ello, se recogen datos de proteínas que intervienen en el proceso de la EA y se estudia que proteínas interaccionan con ellas (PPIs); por otra parte, también se recogen datos de los repositorios de Intact y Negatome sobre proteínas que se tienen pruebas experimentales de que no interaccionan (nPPIs); también, se emparejan proteínas al azar de Uniprot y se asume que son nPPIs. A partir de estas bases de datos, se obtienen las estructuras primarias de las proteínas y se generan características en forma de datos cuantitativos empleando las metodologías de Composición de aminoácidos (AAC), Composición de dipéptidos (DPC), Composición/Transición/Distribución (CTD) y Composición de pseudo-aminoácidos (PAAC). Para elaborar los modelos, a partir de estas características, se emplean los algoritmos Support Vector Machine (SVM) y Random Forest (RF). Finalmente se obtiene que el modelo generado mediante SVM, empleando AAC y empleando la base de datos de Uniprot como fuente de nPPIs es el que mayor capacidad de predicción y robustez presenta.

Abstract (in English, 250 words or less):

Alzheimer's disease (AD) is a neurodegenerative disease that affects a large number of people at this time. Nowadays, numerous therapies are being used to treat it. However, studies on the development of new medications turn out to be expensive processes; therefore, machine learning techniques are being used to reduce costs. In this thesis, machine learning models will be trained to try to predict whether two proteins interact or not. In order to do this, protein data involved in the AD process are collected, and it is then studied which proteins interact with them (PPIs). Data are also collected from the Intact and Negatome repositories on proteins that have experimental evidence showing that they don't have interactions (nPPIs); while random proteins from Uniprot are paired and assumed to be nPPIs. Drawing from these databases, the primary structures of the proteins are obtained and characteristics are generated in the form of quantitative data using the methodologies of Amino Acid Composition (AAC), Dipeptide Composition (DPC), Composition / Transition / Distribution (CTD) and Composition of pseudo-amino acids (PAAC). To develop the models, based on these characteristics, the algorithms Support Vector Machine (SVM) and Random Forest (RF) are used. Ultimately, it is shown that the model generated by SVM, using AAC and using the Uniprot database as a source of nPPIs, is the one with the greatest prediction and robustness.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1 Contexto y justificación del Trabajo

La enfermedad del Alzheimer (EA) es una enfermedad neurodegenerativa que se caracteriza por alteraciones en la memoria, la atención e incluso el lenguaje[1]. Actualmente, la EA es considerada como uno de los mayores problemas sociales y médicos. En el mundo, hay aproximadamente 47 millones de personas afectadas por esta enfermedad [1][2][3]. Según las tendencias detectadas, en los próximos 30 años, esta enfermedad podría afectar a más de 130 millones de personas en el mundo entero[1].

Uno de los principales problemas de la EA es que la enfermedad empieza a desarrollarse hasta décadas antes de que se manifiesten los primeros síntomas, lo cual dificulta enormemente su tratamiento temprano [4][5]. Según la hipótesis amiloide, el proceso de desarrollo de la EA comienza con la acumulación del péptido amiloide-β 1-42 Aβ 1-42), el cual se cree que juega un papel en la muerte celular infiriendo en la conexión entre neuronas durante la sinapsis; además, se produce la fosforilación de la proteína tau, la cual actúa bloqueando el transporte de nutrientes y otras moléculas esenciales para el desarrollo de las neuronas[5][6].

Investigaciones recientes indican que la focalización de la familia de los amiloides para llevar a cabo los tratamientos puede mejorar los resultados clínicos. Sin embargo, el orden y el momento del inicio de la acumulación amiloide y otros procesos de la enfermedad de Alzheimer que conducen a la demencia clínica no se conocen bien [4].

Esta enfermedad, es una de las 10 principales causas de mortalidad en la actualidad, por lo que existe una gran inversión económica sobre los servicios de salud [1]. Esto se ve gravemente afectado debido al envejecimiento de la población.

Se estima que de 2010 a 2050, el número de personas con 70 años o mas que padecerán esta enfermedad aumentará un 153%, esto conlleva un aumento del gasto económico en la inversión de tratamientos que se estima que aumente hasta 1.500 millones de dólares en 2050 [7]. En 2010, el coste anual por cada persona que padecía la EA llego a ser de 73.303 dólares; aunque en el futuro se estima que este coste puede llegar incluso a duplicarse [7].

En Estados Unidos, la EA representa uno de los mayores problemas actuales al ser una enfermedad que aumenta su probabilidad de proliferación con la edad. Esto, unido a que la generación del “Baby Boom” se está acercando a los 65 años, edad en la que el riesgo de padecer la EA incrementa considerablemente, produce una situación alarmante [6].

Debido al aumento esperado de la población con riesgos de padecer la EA, en la Figura 1 se muestra como se prevé que aumenten el número de casos.

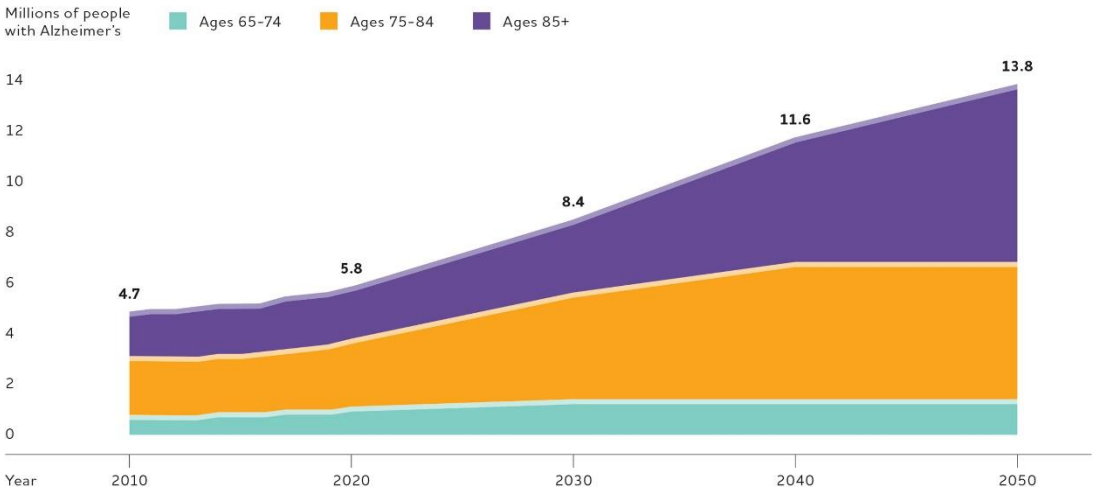

Figura 1: Predicciones del número de personas (en millones) que padecerán la EA entre 2010 y 2050 dividido en rango de edades. Fuente: https://www.alzheimersanddementia.com

Sin embargo, la EA no afecta por igual a todas las personas, sino que existen factores (además de la edad) como el sexo, la raza o el estilo de vida que se relacionan con su desarrollo. Estos factores afectan tanto a la presencia como a la velocidad de desarrollo de esta enfermedad, siendo mayor la incidencia en mujeres [1][8]. Esto no siempre es así, ya que la hormona sexual femenina 17β estradiol, producida durante el periodo menstrual, juega un papel importante en la formación de estructuras como la grasa, los senos o incluso el cerebro, proporcionando protección ante este tipo de degeneración [1]. Tras la menopausia, la producción de esta hormona se ve afectada, por lo que las mujeres contraen un mayor riesgo a la hora de desarrollar la EA.

Siendo una enfermedad tan importante y que afecta a un número tan grande de personas, no es sorprendente que se inviertan una gran cantidad de recursos en su estudio y en técnicas que mejoren su diagnóstico y sus tratamientos.

En la última década, los biomarcadores han atraído una gran cantidad de atención en el diagnóstico clínico. Sin embargo, existen otras enfermedades que pueden alterar el nivel de alguno de los biomarcadores más importantes descubiertos como Aβ 1-42, que se ve reducido en enfermedades como el Párkinson o la demencia vascular [3][9]. Esto complica tanto la diagnosis como el desarrollo de nuevos tratamientos especializados.

A pesar de la dificultad, el lanzamiento de medicamentos con eficacia demostrada se ha convertido en un reto político, económico y mundial debido al amplio mercado existente, sin embargo, la generación de fármacos neurológicos es un proceso muy costoso. Una revisión en 2008 reunió mas de 170 fracasos en el desarrollo de fármacos contra la EA [10].

El desarrollo de fármacos es un proceso complejo de por sí, esto sumado a los problemas y dificultades concretas de las enfermedades neurodegenerativas, complica el proceso considerablemente [10]. En el desarrollo de fármacos, el primer paso es establecer la diana terapéutica, para poder determinar algún compuesto que se pueda unir a dicha diana y así modificar su actividad [11].

Una vez descubiertas estas dianas terapéuticas, el siguiente paso es descubrir compuestos con propiedades farmacológicas que sean atractivas, es decir, baja toxicidad o solubilidad entre otras y sobre todo, que produzca una interacción favorable con la molécula diana [11]. Este estudio se centra en aquellos compuestos de naturaleza proteica que pueden llevar a cabo esta función, es decir, proteínas que interaccionan con otras proteínas (PPI), y que por lo tanto pueden ser potenciales fármacos.

Hasta ahora, se han empleado multitud de técnicas experimentales para estudiar la posible interacción existente entre dos proteínas como ligandos, o si por el contrario, estas proteínas no interaccionan (nPPI), como por ejemplo los sistemas de dos híbridos de levadura (Y2H), la espectrometría de masas (MS), la purificación por afinidad en tándem (TAP) o el chip de proteína [12],[13],[14]. Todas estas técnicas conllevan una gran complejidad tanto desde el punto de vista experimental, como económico; además, cada ensayo cubre un rango muy reducido de interacciones [12].

Por ello, en la actualidad, los esfuerzos se están centrando en desarrollar nuevas técnicas computacionales, la mayoría empleando machine learning y distintas características proteicas, que faciliten la predicción de PPIs a partir de las secuencias primarias de las proteínas [15],[16].

En este trabajo, lo que se pretende es entrenar un modelo que permita reducir los costosos ensayos experimentales para el descubrimiento de PPIs, con el fin de encontrar posibles proteínas con actividad terapéutica para el tratamiento de la EA. Para ello, se emplean como principales herramientas distintos algoritmos de machine learning. El objetivo es obtener distintos modelos, variando el algoritmo de entrenamiento, las características proteicas escogidas y el origen de las nPPIs de las bases de datos para poder comparar y escoger el mejor modelo y la mejor metodología de entrenamiento.

1.2 Objetivos del Trabajo

El principal objetivo de este Trabajo Final de Master es establecer un primer paso para la elaboración de un modelo predictivo que permita dar robustez y fiabilidad a la hora de establecer la presencia o ausencia de interacciones entre proteínas. De esta forma, se pretende reducir la inversión y los tiempos a la hora de establecer candidatos para interaccionar con dianas terapéuticas en el desarrollo de fármacos. De forma más concreta, el modelo se pretende entrenar sobre la EA, empleando proteínas que se relacionan de alguna forma en su desarrollo para entrenarlo.

Para alcanzar el objetivo general del trabajo, se descompone en objetivos mas pequeños que se irán cumpliendo a medida que avanza el estudio:

- Desarrollo de scripts propios para el análisis de los datos.
- Obtener proteínas involucradas en el desarrollo del Alzheimer y proteínas que interaccionan con ellas.
- Elaborar una base de datos en Mongodb con las proteínas que interaccionan y otras que se sabe que no interaccionan.
- Elaborar una base de datos en Mongodb con las proteínas que interaccionan y otras que se eligen aleatoriamente y se considera que no interaccionan.
- Obtener características de diversa naturaleza, a partir de la secuencia primaria de las distintas proteínas.
- Entrenar y evaluar distintos modelos predictivos para estudiar cual es el que mejor que adapta a los datos.

1.3 Enfoque y método seguido

El enfoque que se ha seguido en este estudio es el de obtener una serie de modelos, elaborados con distintos algoritmos de machine learning, para clasificar proteínas según su interacción, estudiando distintas características de su secuencia primaria y el repositorio de procedencia de las proteínas nPPIs. De esta forma, se podría ajustar un modelo de partida para poder llevar a cabo predicciones de interacciones proteicas en proteínas relacionadas en el desarrollo de la EA.

En base a la revisión bibliográfica previa de los estudios recientes se plantearon los siguientes análisis: procesamiento de los datos de grandes repositorios web de proteínas para la obtención del nombre y las secuencias de proteínas de interés, el estudio de la secuencia primaria de las proteínas para obtener características cuantitativas; y una vez generados los modelos, elaborar análisis para poder comparar cuantitativamente la calidad de estos modelos.

Como punto de partida, se tienen los siguientes ficheros:

I. Un fichero con todas las proteínas que se conoce que actúan en el en el desarrollo de la EA [17].

II. Un fichero con PPIs que interaccionan con las proteínas del fichero anterior [18].

III. Un fichero con proteínas nPPIs que tras pruebas experimentales, se ha observado que no interaccionan [18], [19].

IV. Un fichero con proteínas emparejadas al azar, que se interpretan como nPPIs [20].

Para la elaboración del trabajo, se ha utilizado principalmente el lenguaje de programación Python. Python es una herramienta de trabajo muy potente a la hora de trabajar con bases de datos, gracias a librerías como Pandas o Numpy [21], [22]; además, permite hacer una gran cantidad de consultas de tipo biológico [23] y es una potente herramienta de machine learning [24]. Aunque R también es muy potente en este aspecto, se eligió Python porque contiene librerías mejor conocidas, ya que casi todas fueron específicamente introducidas en la asignatura de Programación para la bioinformática de este master.

Para la generación de los características cuantitativos a partir de la estructura primaria de las distintas proteínas se empleó el servidor web iFeatures, el cual proporciona una gran cantidad de sistemas para obtener características proteicas a partir de su secuencia de aminoácidos [25].

Para almacenar la información generada durante el trabajo, se emplea una base de datos no relacional, Mongodb en nuestro caso. Se emplea este tipo de bases de datos ya que en este caso no se necesitan relacionar

las distintas tablas; además la estructura de las bases de datos no está muy clara por lo que esta es la mejor forma de almacenamiento [26].

1.4 Planificación del Trabajo

Los principales recursos que se emplean en este trabajo son los distintos ficheros que contienen la información de las proteínas de interés que se han mencionado en el apartado anterior. Para su manipulación se han elaborado una serie de Scripts propios con Python. Dichos ficheros de partida y los scripts desarrollados para manipularlos se encuentran en repositorio GitHub (https://github.com/cperezlopez/TFM-bioinform-tica) [27].

Una vez se han obtenido los distintos ficheros, hay que conseguir el nombre de las distintas proteínas para poder separarlas según se vaya a trabajar con ellas como PPI o como nPPI. Tras tener separadas las proteínas, se elabora un script de Python para poder extraer las secuencias de aminoácidos para cada una de las proteínas.

A partir de estas secuencias proteicas, se emplea iFeatures [25] para lograr una serie de características que se puedan medir cuantitativamente para poder compararlas. A partir de estas características obtenidas de las distintas proteínas, se crean distintas colecciones en Mongodb, para almacenar los resultados.

Finalmente, se emplean estas colecciones para generar una serie de modelos predictivos de los cuales se extraen una serie de parámetros para poder compararlos entre sí y poder evaluar cual es la mejor metodología para estudiar las PPI o nPPI de la EA.

De forma más esquemática, estas tareas se pueden establecer:

- Crear scripts propios en Python para realizar el resto de las tareas.
- Obtener una lista con las proteínas relacionadas en el proceso de la EA.
- Obtener tres bases de datos, la primera con las proteínas anteriores y una de sus PPI, otra con proteínas que se sabe que son nPPIs y una última con proteínas emparejadas al azar y que se establece que son nPPI.
- A partir de las secuencias de las proteínas, obtener las distintas características, para poder hacer las secuencias comparables.
- Elaborar los modelos predictivos empleando Support Vector Machine (SVM) y Random Forest (RF) [13],[15],[14].
- Con los datos de evaluación, comparar los modelos aplicando distintas medidas: Exactitud (ACC), Precisión (PE), Sensibilidad (SEN), Especificidad (SPE), Coeficiente de correlación de matthews (MCC) y Puntuación F1 [12],[13],[14],[28].

Antes de llevar a cabo cualquiera de estas tareas, hay que realizar la propuesta y la planificación del Trabajo final de Master. Esta propuesta se hace a modo de 2 Pruebas de Evaluación Continua (PEC), las cuales pertenecen a las PEC 0 y PEC 1 de la asignatura, donde se realiza una búsqueda bibliográfica inicial para estimar las distintas etapas del trabajo y el tiempo aproximado que puede llevar cada una de estas.

Durante el desarrollo del trabajo, se realizan otras 2 PECs (PEC 2 y PEC 3, las cuales, junto a la PEC 1 se añadirán como anexos del trabajo). En estas otras PECs se realiza un seguimiento del trabajo y se estudia el correcto cumplimiento de las etapas establecidas y de los distintos periodos planificados para cada tarea establecida en la PEC 1. Además de todo esto, también se analizan los distintos contratiempos, cambios o retrasos que hayan podido surgir durante la elaboración de las PECs.

Para concluir este trabajo, se prepara una memoria final y se defiende ante un tribunal mediante una presentación en vídeo.

A continuación, se muestra la elaboración de un esquema que resume las actividades realizadas durante todo el periodo de desarrollo del Tarbajo Final de Master.

- Septiembre 2019:
  - PEC0: Definición de los contenidos del trabajo.
- Octubre 2019:
  - PEC1: Plan de trabajo.
  - Generar una base de datos con proteínas relacionadas con la EA y PPIs, otra con proteínas nPPIs y otra con proteínas unidas al azar.
- Novembre 2019:
  - PEC2 - Desarrollo del trabajo Fase I.
  - Obtener las secuencias de todas estas proteínas e introducirlas en Mongodb.
  - Obtener características a partir de la estructura primaria.
- Diciembre 2019:
  - PEC3 - Desarrollo del trabajo Fase II.
  - Unir las bases de datos en 2: PPIs +nPPIs y PPIs + unidas al azar
  - Generar y evaluar los modelos.
- Enero 2019:

- PEC4: Entrega de la memoria final.
- PEC5a: Elaboración de la presentación
- PEC5b: Defensa pública

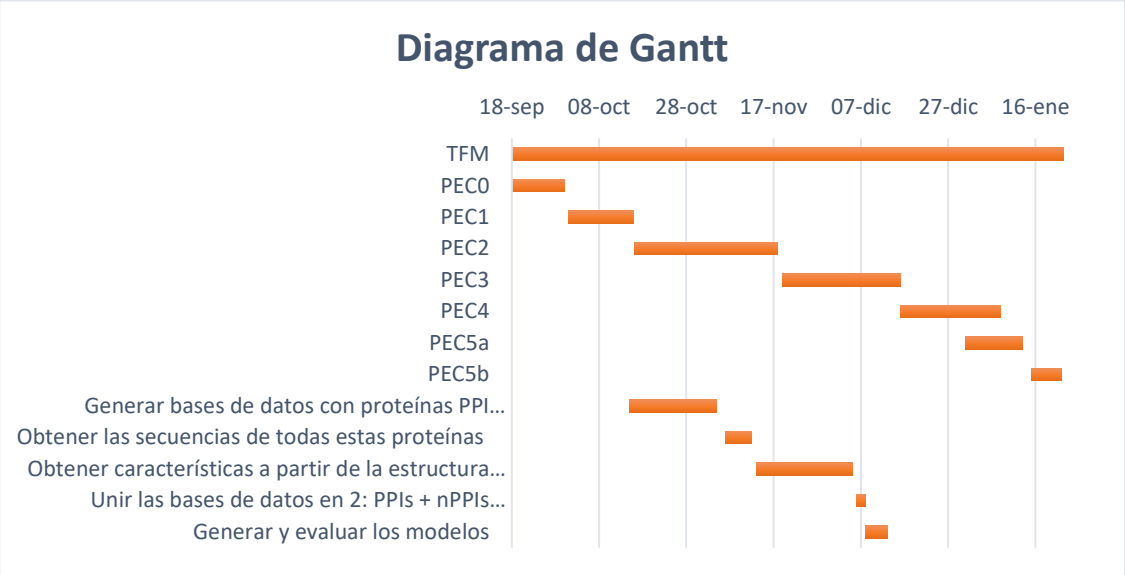

Figura 2: Diagrama de Gantt. Resumen de las distintas PECs y pasos a realizar para llevar a cabo el trabajo.

1.5 Breve sumario de productos obtenidos

En este trabajo se han realizado diversas tareas, por lo que se han generado una gran cantidad de archivos, estos archivos se pueden dividir en dos apartados:

- Archivos que se entregan para la evaluación por parte de la Universitat Oberta de Catalunya:
  - Las distintas PECs: PEC0, PEC1, PEC2 y PEC3.
  - La Memoria Final del Trabajo.
- Archivos generados de la elaboración del trabajo:
  - Los distintos Scripts generados con Python, almacenados en el repositorio de GitHub
  - Los distintos ficheros generados del procesamiento de las bases de datos, también en el repositorio de GitHub.
  - Los resultados del entrenamiento y evaluación de los distintos modelos.

1.6 Breve descripción de los otros capítulos de la memoria

Una vez se ha llevado a cabo una introducción en la que se ha comentado un poco el objetivo del estudio, la problemática a resolver y la planificación temporal del mismo; hay que proceder a desarrollar el trabajo con mayor profundidad. Esto se va a realizar a lo largo de los distintos capítulos:

- Materiales y métodos: en este capítulo se van a explicar los distintos pasos que se han realizado y los razonamientos que han llevado a las distintas tomas de decisiones. Además, se realizará una descripción de los distintos archivos de partida.

- Resultados y conclusiones: en este capítulo, por un lado, se expondrán los distintos resultados que se obtienen de cada una de los pasos que se realizan en el capítulo de materiales y métodos, así como los resultados finales. Por otra parte, también se comentarán las distintas conclusiones que se obtienen de estos resultados, es decir, se analizan las distintas conclusiones que se pueden sacar de estos resultados. Por último, se analizarán las conclusiones de los resultados finales del proyecto.

2. Materiales y métodos

2.1 Datos

En este apartado, se habla de los distintos datos de partida que se emplean en este trabajo, así como las variables que contienen la distinta información de interés.

Este proyecto comienza con datos de proteínas de 5 repositorios diferentes. Unos datos contienen información sobre las distintas proteínas que tienen referencia de intervenir de alguna manera en el proceso de desarrollo de la EA [17], otro posee información de proteínas que tienen algún tipo de interacción demostrada empíricamente con el fichero anterior, otros dos poseen proteínas que no tienen ningún tipo de interacción demostrado empíricamente[18][19] y por último, un fichero con proteínas unidas al azar, las cuales se considera que tampoco tienen ningún tipo de interacción [20].

2.1.1 Proteínas relacionadas con el Alzheimer y sus interacciones

Los datos de este estudio han sido generados a partir de bases de datos de uso público. Para obtener las distintas proteínas, que de una forma u otra, tienen alguna función en el desarrollo de la EA, se accedió a los datos de Disgenet [29], un repositorio de bases de datos donde se almacenan proteínas relacionadas con distintas enfermedades.

El archivo descargado de esta página web, contiene una gran cantidad de información: nombre e identificador (id) de la enfermedad, nombre e id de la proteína, referencia en Uniprot, clase de proteína, número de enfermedades con las que se ha relacionado, una serie índices de asociación a la enfermedad en cuestión, cantidad de single nucleotide polymorphism, el año en el que consta la primera referencia de la proteína y el año en el que se publicó la última referencia.

De toda esta información, los datos que interesan obtener de este fichero son los de la columna que contiene el nombre de referencia en Uniprot para posteriormente poder obtener la secuencia de aminoácidos de las distintas proteínas.

Otro de los ficheros de los cuales parte este estudio es el fichero de interacciones de proteínas demostradas de forma experimental. Para obtener este fichero se accede a la base de datos de IntAct [30]. En esta base de datos, se descarga un fichero que contiene los nombres de proteína A y B (las cuales tienen registro experimental de interacción), los métodos de interacción, información de los estudios donde se demostró la interacción, tipo de interacción, función biológica de cada una y una serie de características que no están relacionadas con este estudio.

Para este estudio lo que se pretende es buscar las distintas proteínas que se han obtenido relacionadas con la EA y ver si existen interacciones conocidas. Como se puede prever, para un gran número de proteínas de estudio, existen una gran cantidad de interacciones conocidas, sin embargo, para este estudio, se selecciona aleatoriamente únicamente una de estas PPIs para evitar redundancia en los datos ya que existe un gran número de proteínas PPI; de esta forma se pueden representar todas ellas; sin perder las proporciones que se comentarán más adelante.

Finalmente, se obtienen un total de 1684 parejas de proteínas PPIs, de las cuales, una de las proteínas que conforman la pareja, esta relacionada en el desarrollo de la EA.

2.1.2 Proteínas sin interacciones demostradas

Para la obtención de proteínas que estuviese documentado experimentalmente que carecen de cualquier tipo de interacción se emplearon dos bases de datos pertenecientes a dos repositorios web diferentes.

Por un lado, se descargó un fichero de la página de IntAct [30]. Este fichero tiene las mismas características que el fichero que se descargó para el apartado anterior, la única diferencia es que las dos primeras columnas, antes pertenecían a los ids de proteínas entre las que se establece alguna interacción; mientras que, en este caso, estas dos primeras columnas se corresponden a ids de proteínas nPPI.

Finalmente, se obtienen 430 parejas de proteínas con evidencias experimentales de que no interaccionan entre sí a partir de esta base de datos.

Por otro lado, se descargó otro fichero de proteínas con la intención de obtener un mayor número de nPPIs. Se accedió al repositorio de Negatome [31] para obtener esta información.

En este caso, el fichero que se obtiene de este repositorio es mucho mas sencillo, consta únicamente de 4 columnas, dos columnas con los nombres de las nPPIs, otra con su id en PubMed y, por último, una columna con la prueba experimental empleada que demuestra la falta de interacción proteica. Siguiendo las pautas anteriores, lo único que interesa es aislar los nombres de las proteínas, es decir las 2 primeras columnas.

Como consecuencia se obtiene un fichero con 1020 proteínas con que se tiene constancia experimental de que no interaccionan, según la información que ha recogido este proyecto [19].

2.1.3 Proteínas sin interacciones consideradas

El objetivo de esta fase del trabajo es elaborar una base de datos a partir de la cual entrenar un modelo, por lo tanto, es importante que esta

contenga diferenciadas las PPIs de las nPPIs para una mayor eficiencia en el posterior entrenamiento [12].

Debido a la importancia de la calidad de las bases de datos, se va a crear otra base de datos de nPPIs para poder comparar las anteriores por si existiera algún tipo de sesgo [19]. Otra forma de obtener una base de datos de nPPI es emparejando proteínas al azar y considerando que entre estas proteínas no existe ninguna interacción [19] [32].

Por ello, se descarga la base de datos de Uniprotkb cruzada con SwissProt que proporciona Uniprotkb [33]. Este archivo proporciona la información de referencia en Protein Data Bank (PDB), información sobre la metodología de identificación de la proteína, el nombre que recibe en SwissProt y el código en Uniprotkb. Como en el resto de los casos, lo que interesa del fichero es esta última columna del nombre que recibe en Uniprotkb.

Una vez aislado el nombre, se seleccionan una serie de proteínas al azar y se emparejan asumiendo que, debido a la baja probabilidad que existe, estas no interaccionan [32]. Finalmente, se obtiene una base de datos con 1450 pares de proteínas que se asume que no interaccionan.

2.1.4 Bases de datos para entrenar el modelo

Tras la obtención del nombre de las distintas proteínas que conforman cada una de las categorías, hay que conseguir una serie de información cuantitativa, que permita comparar las distintas parejas de proteínas, para poder entrenar al modelo de clasificación.

Lo primero que se necesita es la estructura primaria de las proteínas, es decir, la secuencia de aminoácidos que se asocia a cada nombre de las proteínas. Para ello, se realizan consultas a través de GeneBank [34]. Estas consultas producen una serie de errores gramaticales con las distintas proteínas, a la hora de hacer la consulta.

Para solucionar estos errores, se eliminan aquellas proteínas que producen el error. En primer lugar, se eliminan aquellas con el carácter ‘-‘ y se guardan las proteínas para que los ids se ajusten de nuevo y para que no haya posiciones inexistentes. A continuación, se eliminan el resto de proteínas que producen errores y se vuelven a guardar para, nuevamente, reestablecer los ids para las consultas.

Tras este paso, las 2 bases de datos obtenidas de Intact y Negatome se ven algo reducidas, por lo tanto, y dada la similitud de su naturaleza se juntan en una sola base de datos, obteniendo así 3 bases de datos. De la misma forma, la base de datos de Uniprot también se ve reducida, aunque no de forma tan drástica como Intact y Negatome.

Por lo tanto, como resultado se han conseguido identificar las estucturas primarias de 1570 parejas de PPI, 1031 parejas de nPPI provenientes de Intact y de Negatome y 1101 parejas de nPPI obtenidas de Uniprotkb.

Una vez asociados a los nombres de las proteínas su secuencia de aminoácidos correspondiente, se pueden obtener una serie de características cuantitativas que permitan comparar las estructuras primarias de las proteínas y así poder entrenar los modelos de clasificación. Para obtener estas características, se emplean distintas metodologías, de esta forma además de comparar las distintas bases de datos de donde se han obtenido las nPPI, también se podrá evaluar que metodología para la obtención de características a partir de la secuencia de aminoácidos se ajusta mejor.

Antes de comenzar con las consultas, se escogen las proteínas definitivas que se van a emplear. Dado que las PPI no se pueden tocar ya que, son las proteínas de interés y deben permanecer intactas; y que no se pueden añadir mas proteínas para lograr una proporción 1:1 a la base de datos conformada con las proteínas de Intact mas las de Negatome, se eliminarán aleatoriamente proteínas de esta base de datos y de la de Uniprotkb para lograr una proporción de 1:2. De esta forma se eliminan proteínas al azar, hasta que quedan 785 en cada una de las bases de datos que contienen nPPIs.

Para la obtención de las características cuantitativas se emplearon distintas metodologías [25].

2.1.4.1 Composición de aminoácidos (AAC)

La composición de aminoácidos es una metodología muy sencilla y una de las mas empleadas para la obtención de características de las proteínas [35]. Consiste en la obtención de 20 valores, cada uno de los cuales se corresponde con la frecuencia que tiene cada uno de los aminoácidos dentro de la estructura primaria de la proteína [36]. La fórmula 1 representa como obtener las distintas frecuencias.

$$f(t) = \frac{\mathrm{N(t)}}{\mathrm{N}} \quad (1)$$

Donde f(t) representa la frecuencia de un aminoácido, N(t) las veces que aparece ese aminoácido en la proteína y N el número total de aminoácidos que tiene la proteína.

2.1.4.2 Composición de dipéptidos (DPC)

Uno de los principales problemas que puede tener AAC es que no tiene en cuenta la posición que ocupan los distintos aminoácidos [36]. La composición de dipéptidos proporciona la frecuencia en la que se da cada dipéptido, de esta forma se obtienen 400 descriptores [37]. La fórmula 2 representa como obtener cada uno de estos descriptores.

$$D(r,s) = \frac{\mathrm{N(rs)}}{\mathrm{N}-1} \quad (2)$$

Donde N(rs) es el número de dipéptidos compuestos por los aminoácidos r y s.

2.1.4.3 Composición/Transición/Distribución (CTD)

Fue presentada para intentar predecir las distintas clase de plegamiento de las proteínas [38]. Después se empezó a aplicar en algoritmos de clasificación según la secuencia de aminoácidos [39]. Representa el patrón de distribución de los distintos aminoácidos a lo largo de la secuencia primaria, en función de sus propiedades físico-químicas. Esta metodología emplea siete propiedades fisicoquímicas, que incluyen hidrofobicidad, polarización, volumen normalizado de Van Der Waals, estructura secundaria, polaridad, carga y accesibilidad a solventes.

Todos los aminoácidos que forman el péptido se dividen en tres grupos: polares, neutros e hidrófobos. La C representa tres valores de representación porcentual para un péptido dado: polar, neutro e hidrófobo. La T, representa el porcentaje de frecuencia de un aminoácido polar seguido de uno neutro o de uno neutro seguido de uno polar; también puede representar el porcentaje de uno hidrófobo seguido de uno neutro o uno neutro seguido de uno hidrófobo; así como uno polar seguido de uno hidrófobo o uno hidrófobo seguido de uno neutro. Y, por último, D tiene hasta cinco descriptores dentro de cada grupo; mide la longitud de la cadena dentro de la cual se encuentran los primeros 25, 50, 75 y 100% de los aminoácidos de una propiedad específica [35],[37],[39].

2.1.4.4 Composición de pseudo-aminoácidos (PAAC)

El concepto de PAAC se desarrolló originalmente para predecir localizaciones subcelulares de proteínas [40]. La idea básica del PAAC es extraer información, no solo de los aminoácidos de la secuencia, sino también, del orden que tienen con los coeficientes de autocorrelación de la secuencia de la proteína si cada residuo en la secuencia de la proteína puede representarse con un número [41]. Se ha demostrado que es muy efectiva en muchas proteínas y sistemas basados en proteínas [36]. El PAAC, para cada secuencia de proteína se puede representar mediante un vector (20 + d) -dimensional. Información más detallada de esta compleja metodología se puede consultar en la siguiente referencia [25].

2.1.5 Bases de datos para entrenar los modelos

Una vez realizadas las consultas con las secuencias de cada una de las bases de datos, se unen las tres bases de datos para entrenar los modelos. Por una parte, se genera una base de datos que tiene las PPI y las nPPIs obtenidas de Intact y Negatome; por otro lado, se genera otra base de datos con las PPI y las nPPIs de Uniprotkb.

Finalmente se han obtenido dos bases de datos, de las cuales se realizan cuatro consultas, una para cada metodología descrita en el apartado anterior. De esta forma, se tendrán 8 bases de datos que permitirán comparar las distintas metodologías de obtención de características a partir de la secuencia de aminoácidos; y las bases de datos empleadas para obtener las nPPIs.

2.2 Software empleado

2.2.1 Mongodb

Para almacenar los datos después ser tratados o después de realizar las distintas consultas, se emplea la base de datos no relacional Mongodb. Mongodb es una base de datos abierta que proporciona una gran escalabilidad, lo que hace que sea una base de datos muy empleada. Por lo tanto, una gran cantidad de datos biológicos son almacenados en Mongodb ya que proporciona un formato simple, sobre todo cuando se trabaja con datos no estructurados [42].

De esta forma, las diferentes bases de datos obtenidas tras realizar alguna consulta serán almacenadas en Mongodb. Por una parte, se almacenarán los datos obtenidos de una primera consulta donde, a partir de los nombres de las proteínas, se obtiene su secuencia de aminoácidos; por otra parte, también se almacenarán los datos obtenidos de las distintas consultas para obtener características a partir de la estructura primaria de las proteínas.

2.2.2 Python

Para este estudio, se ha empleado como principal herramienta de trabajo el lenguaje de programación Python para llevar a cabo el tratamiento de las distintas bases de datos [43].

Python es un lenguaje interactivo y orientado a objetos, cuya sintaxis es sencilla e intuitiva, está basado en códigos de alto rendimiento como Fortran, C o C++ [44]. Este lenguaje se ha convertido en uno de los lenguajes mas populares en el manejo de datos debido a la robustez y calidad de sus librerías especializadas en estas funciones [45]. Para crear los scripts empleados en el tratamiento de los datos, se emplea Júpiter Notebook, un entorno de trabajo basado en el navegador que permite la integración perfecta de código, texto narrativo y figuras en un solo elemento ejecutable, que se puede editar instantáneamente. Además, los cuadernos de Jupyter pueden reconocer varios lenguajes de programación, por lo que se puede escribir un cuaderno combinando varios lenguajes [45].

Además de las distintas funciones preprogramadas que proporciona Python, existe la posibilidad de habilitar paquetes de extensión. En este trabajo, se han empleado varias funciones que se almacenan en estos paquetes de extensión. Principalmente, se han empleado las librerías Pandas, Biopython, Sklearn y PyMongo.

La librería Pandas [46] es una librería de alto rendimiento que posee funciones para el tratamiento de tablas con una gran cantidad de datos [45]. En sus comienzos, esta librería se empleaba para gestionar bases de datos orientadas a finanzas, sin embargo, su uso se ha visto cada vez mas inclinado a temas académicos e industriales [22].

El proyecto de Biopython [47] es un proyecto internacional de código abierto desarrollado por voluntarios que proporciona una serie de herramientas de Python para consultas bioinformáticas [23]. En este caso la consulta de mayor uso ha sido la función Entrez, empleada para la obtención de las secuencias de aminoácidos de las proteínas [34].

La librería Sklearn [48] es una librería de Python con una serie de funciones que proporcionan un amplio rango de algoritmos de machine learning, tanto supervisados como no supervisados [44].

PyMongo [49] es una librería que permite la creación de bases de datos a partir de diversas funciones. De esta forma, se puede introducir, consultar y extraer información de la base de datos de forma sencilla a través de código Python.

Todos los scripts que se han desarrollado a lo largo del trabajo, se han depositado en un repositorio público de libre acceso llamado GitHub. Esta plataforma es muy conocida y permite depositar trabajos y scripts de forma pública o privada.

La finalidad de este repositorio es facilitar la corrección de los scripts que se elaboraron para el procesamiento de ficheros y las consultas realizadas durante este trabajo.

Link de acceso: https://github.com/cperezlopez/TFM-bioinform-tica

2.2.3 iFeatures

La herramienta iFeatures es un servidor de consultas que permite obtener distintas características físico-químicas de las proteínas y péptidos a partir de su estructura primaria, siendo una herramienta ampliamente utilizada para la predicción de estructuras, funciones e interacciones de proteínas y péptidos [25].

Es una herramienta versátil de Python que permite emplear 18 metodologías para obtener hasta un total de 53 descriptores a partir de la estructura primaria de las proteínas. Además, este servidor integra 12 tipos diferentes de algoritmos de agrupación de características, selección y reducción de dimensionalidad comúnmente utilizados [25].

Dentro de la gran cantidad de metodologías mencionadas arriba, en este trabajo se realizan consultas para Composición de aminoácidos, Composición de dipéptidos, Composición/Transición/Distribución y Composición de pseudo-aminoácidos.

2.3 Algoritmos

En este trabajo se emplean algoritmos de machine learning para crear los modelos de clasificación de pares de proteínas. A cada una de las 4 bases de datos creadas hasta ahora, se le aplican dos algoritmos: Support Vector Machine y Random Forest [13],[14],[15],[36]. Tras la elaboración de los distintos modelos, se lleva a cabo una comparación de los resultados para evaluar cuál de los modelos se ajusta mejor a los datos; obteniendo así el mejor algoritmo, metodología de extracción de propiedades a partir de la secuencia primaria y base de datos nPPI combinadas.

2.3.1 Support Vector Machine

El algoritmo SVM ha sido empleado con éxito en diversos estudios biológicos, incluyendo estudios bioinformáticos y de biología computacional [35]. El objetivo de este algoritmo es crear un límite de decisión entre dos clases, conocido como hiperplano, que permita la predicción de etiquetas a partir de uno o varios vectores de características. Esta separación está orientada de tal forma que se aleje lo máximo posible de los puntos mas cercanos, a los que se les denomina soportes [50].

En su caso más simple, los datos que se pretenden separar se encuentran en una única dimensión y son separables linealmente, como se observa en la Figura 3.

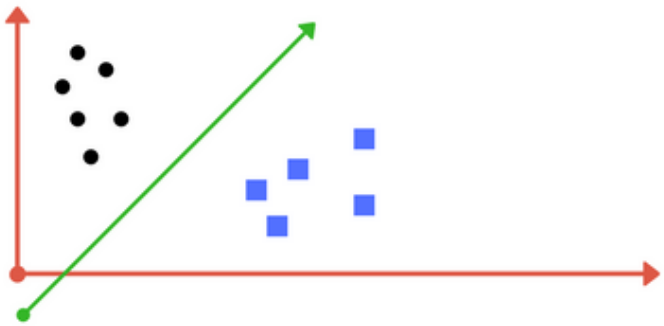

Figura 3: Separación de los dos grupos de datos mediante un hiperplano. Fuente: https://medium.com

Aunque pueden existir distintos hiperplanos que separen correctamente los datos, se busca el máximo margen de hiperplano, en el que el hiperplano establece la mayor separación posible entre las dos clases de datos [51].

En ciertos casos, los datos puede que no sean separables simplemente a partir de una línea recta. Existen varias soluciones posibles a este tipo de problemática, una de estas soluciones es permitir que alguno de los puntos, caiga en el lado incorrecto del margen [51].

Sin embargo, no siempre esta es la mejor opción, ya que existen casos en el que el error aceptado sería demasiado elevado (Figura 4). Una de las características clave del SMV es la capacidad de añadir una nueva dimensión para intentar aportar una solución a este problema, mediante un proceso denominado kernel trick (Figura 5) [51].

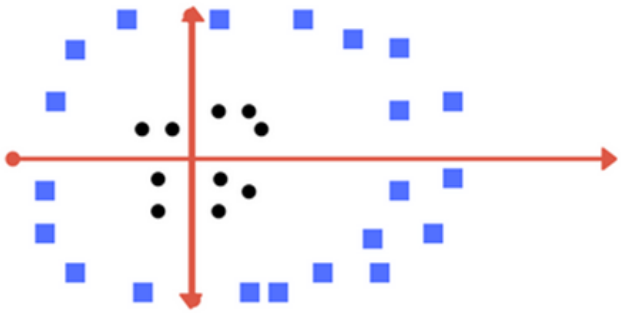

Figura 4: Disposición de datos no separables en dos dimensiones de forma lineal. Fuente: https://medium.com

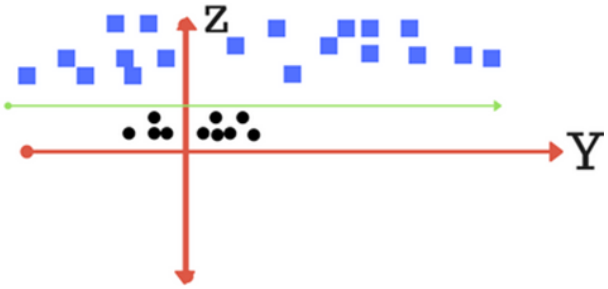

Figura 5: Separación de los datos tras añadir la nueva dimensión Z mediante kernel trick. Fuente: https://medium.com

Existen distintos tipos de kernels, aplicables a diferentes tipos de datos. Algunos de los kernels mas comúnmente utilizados son el lineal, el polinómico, el sigmoide y el radial (RBF) [52].

Además de establecer un kernel correcto, existen otros parámetros que son necesarios establecer para ajustar el mejor modelo posible. El parámetro de regularización (parámetro C) indica cuanto se desea evitar perder en la clasificación de cada ejemplo de entrenamiento; cuanto mayor valor tiene la C el margen del hiperplano tendrá un tamaño menor [52].

Por otra parte, el parámetro Gamma, define hasta donde llega la importancia de un único ejemplo de entrenamiento, es decir, que cuanto mayor es este valor, se le otorgará mayor importancia a los puntos que se encuentren mas próximos al hiperplano [52].

Para establecer cuales serían los valore elegidos para cada uno de los distintos parámetros, habría que llevar a cabo un análisis exhaustivo en el que se van acotando las distintas opciones hasta encontrar la óptima. Sin embargo, teniendo en cuenta las limitaciones de este estudio, se escogen una serie de valores para cada uno de los parámetros y se

emplea la función GridSearchCV [53] la cual emplea el método cross-validation, para determinar cuales son los parámetros que mejor se ajustan y cual es el mejor modelo de entrenamiento que se puede establecer empleando una k = 10 [54].

2.3.2 Random Forest

El algoritmo RF es uno de los algoritmos de clasificación más empleados. Utiliza una colección de árboles de decisiones para reducir la varianza de estos arboles y mejorar el poder predictivo de los modelos. Este algoritmo es uno de los mas empleados actualmente en machine learning [14].

Para entender el algoritmo de RF, hay que comprender el algoritmo de árboles de decisiones, ya que se trata de un conjunto de cientos o miles de estos árboles independientes [39].

La estructura de un árbol de decisiones comienza con un nodo central que se va dispersando y va dando lugar a los distintos nodos de decisión, estos nodos de decisión se basan en atributos o características del elemento que se quiere clasificar. Cada uno de los nodos desembocan en distintas ramas, las cuales representan los resultados de los nodos anteriores. Esta estructura continúa hasta que se alcanzan los nodos terminales donde se clasifica el elemento estudiado [55],[51].

Su nombre es debido a esta estructura en forma de árbol que se ramifica hasta poder clasificar los distinto elementos como se observa en el ejemplo de la Figura 6, sobre condiciones climatológicas para practicar deporte o no.

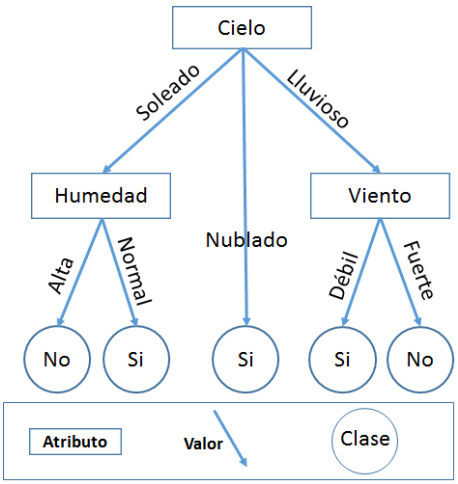

Figura 6: Ejemplo de un esquema de un árbol de decisiones en el que se estudia las condiciones climatológicas para practicar deporte. Fuente: http://numerentur.org

Teniendo en cuenta esto, el algoritmo de RF está compuesto de diferentes árboles de decisión, cada uno con los mismos nodos, pero los datos que se emplean en ellos son distintos y conducen a hojas diferentes. De esta forma, tiene en cuenta las decisiones de múltiples árboles para encontrar una respuesta, que resultará del promedio de todos los árboles de decisiones estudiados [56].

Igual que en algoritmo de SVM, existen unos parámetros que se deben ajustar para obtener la máxima rentabilidad. El parámetro que se ajusta en este caso es el número de árboles empleados, este valor, a priori mejora cuanto mayor es el número de árboles empleados. Sin embargo, el coste computacional aumenta considerablemente para entrenar el modelo, sin que se obtenga una mejora a tener en cuenta de los resultados del modelo [56].

Igual que en el algoritmo de SVM, se estudia cual puede ser el mejor valor para este estimador, dentro de una serie de valores que cogen un amplio rango. Para ello, se emplea la función GridSearchCV [53]. Esta función emplea el método cross-validation para determinar cuáles es el parámetro que mejor se ajusta y cuál es el mejor modelo de entrenamiento que se puede establecer empleando una k = 10 como valor para el análisis cross-validation [54].

2.3.3 Evaluación

Para elegir el modelo que mejor se ajusta al estudio, hay que establecer una serie de metodologías de evaluación que permitan comparar las características de los distintos modelos para detectar las fortalezas y debilidades de cada modelo respecto al resto.

Los parámetros empleados en este estudio para evaluar los distintos modelos son la exactitud (ACC), precisión (PE), sensibilidad (SEN), especificidad (ESP), coeficiente de correlación de Matthews (MCC) y la puntuación F1 (F1) [12],[37],[39],[57].

Los distintos métodos, utilizan los resultados de la evaluación individual de cada uno de los modelos para obtener sus valores, es decir, emplean los verdaderos positivos (TP), son los valores que se reconocen de forma correctos como PPI; los verdaderos negativos (TN), son los valores que se reconocen correctamente como nPPI; los falsos positivos (FP), son valores que el modelo clasifica como como PPI siendo nPPI; y los falsos negativos (FN), se corresponden con casos en los que el modelo clasifica como nPPI algún valor que se corresponde con PPI [51]. Todos estos valores se obtienen analizando los resultados tras probar el modelo con los valores de evaluación y comparando los resultados de las predicciones con los valores reales.

Para calcular los distintos parámetros de evaluación se emplean las siguientes fórmulas:

$$\mathrm{ACC} = \frac{TP+TN}{\mathrm{TP+TN+FP+FN}} \tag{3}$$

$$\mathrm{SEN} = \frac{TP}{\mathrm{TP+FN}} \tag{4}$$

$$\mathrm{SPE} = \frac{TN}{\mathrm{TN+FP}} \tag{5}$$

$$\mathrm{PE} = \frac{TP}{\mathrm{TP+FP}} \tag{6}$$

$$\mathrm{MCC} = \frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (7)$$

$$\mathrm{F1} = \frac{2TP}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \quad (8)$$

Todos estos valores serán calculados para cada uno de los modelos y se recogen en una tabla para poder comparar cuantitativamente los modelos generados.

3. Resultados y discusión

En este apartado, se van a analizar los pasos realizados anteriormente y se van a interpretar los resultados para poder determinar cual es la metodología, de entre todas las empleadas, que mejor se ajusta a los datos con los que se trabaja.

3.1 Parámetros de los algoritmos

El objetivo de esta primera parte es determinar, de entre una serie de valores para los distintos parámetros de los algoritmos, cuáles son los que mejor se ajustan para el entrenamiento de los modelos en cada uno de los casos.

Se emplea la función GridSearchCV para analizar cuál es la mejor combinación de parámetros que se pueden emplear en el entrenamiento de los modelos. Además, se emplea el método k-fold cross validation para determinar como se deben distribuir los datos de forma óptima para llevar a cabo este entrenamiento.

3.1.1 SVM

Una vez estudiados una serie de parámetros como candidatos para generar cada uno de los modelos, en la Tabla 1 se muestran aquellas combinaciones que se han obtenido como las mejores para el algoritmo SVM, en cada uno de los casos. Como se ha explicado, existen muchas combinaciones posibles, sin embargo, este trabajo conlleva unas limitaciones temporales y computacionales, por lo que se reunirán unos valores y se estudiarán los mejores dentro de estos valores proporcionados. Estos parámetros solo se han aceptado cuando se descartan valores mayores y menores, es decir, cuando en el rango proporcionado hay valores más grandes y más pequeños que han sido descartados. De esta forma, se evita que se escoja este valor siendo que hay otro más adecuado fuera del rango de los valores estudiados.

Tabla 1: Parámetros empleados para entrenar el modelo en SVM para cada caso. Estos parámetros se han escogido como los mejores, de entre un rango de parámetros proporcionado, para llevar a cabo el entrenamiento.

Database	Features	C Parameter	Gamma
Uniprot	AAC	50000	0.1
Intact/Negatome	ACC	1000	1
Uniprot	DPC	100	0.5
Intact/Negatome	DPC	50000	0.001
Uniprot	CTD	100	1,00E-05
Intact/Negatome	CTD	5	0.001
Uniprot	PAAC	10	0.001
Intact/Negatome	PAAC	1	0.01

En cuanto a las metodologías de la obtención de las características a partir de la estructura primaria de las proteínas, de observa que aquellas con un procedimiento mas sencillo y cuyos parámetros son más simples [25], tienen valores generalmente superiores a las metodologías con parámetros más complejos; tanto para el parámetro C, como para el parámetro gamma.

Esto representa que cuanto mayor es la complejidad de la metodología para la obtención de las características, se requiere unas restricciones mas laxas para obtener un mayor margen del hiperplano [58]; además de una menor relevancia para los ejemplos individuales de entrenamiento para obtener el modelo de clasificación más eficiente [52],[59].

Para cada metodología de obtención de las características, los valores recogidos para el parámetro C es mayor en los casos en los que la base de datos empleada para obtener las nPPIs se corresponde con Uniprot. Esto se observa en todos los casos, a excepción de DPC.

3.1.1.2 RF

Tras estudiar el parámetro del número de árboles que se van a emplear en el algoritmo RF para cada caso (parámetro n_estimators en el algoritmo empleado en el script de Python), en la Tabla 2 se reflejan aquellos valores que se consideran los mejores para entrenar cada uno de los modelos.

Igual que en apartado anterior, en este caso también se han seleccionado aquellos que, dentro del rango estudiado, ha habido valores que se han descartado siendo mayores y menores que el escogido. Así se asegura que, el valor escogido se acerque en la medida de lo posible al valor óptimo y no quede fuera de este rango de valores que se han estudiado.

Tabla 2: Parámetros empleados para entrenar el modelo RF para cada uno de los casos. Estos parámetros se han escogido como los mejores, de entre un rango de parámetros proporcionado, para llevar a cabo el entrenamiento.

Database	Features	N_estimators
Uniprot	AAC	100
Intact/Negatome	ACC	500
Uniprot	DPC	100
Intact/Negatome	DPC	250
Uniprot	CTD	50
Intact/Negatome	CTD	300
Uniprot	PAAC	300
Intact/Negatome	PAAC	400

En la Tabla 2, se observa que el número de árboles empleados en cada una de las metodologías de obtención de características a partir de la estructura primaria de las proteínas, sigue el mismo patrón en todas. En las distintas metodologías, el número de árboles escogido como valor óptimo es mayor para las bases de datos donde las nPPI se han

obtenido de la suma de Intact y Negatome, respecto a las bases de datos de Uniprot.

Estos parámetros indican un mayor coste computacional, debido a la mayor cantidad de árboles necesarios para elaborar el modelo en las bases de datos donde las nPPIs vienen de Intact y Negatome [60]. Esto se debe a la naturaleza de los datos obtenidos en Intact y Negatome, los cuales, pueden arrastrar algún sesgo [19].

A la hora de escoger el número de árboles, se busca encontrar el equilibrio entre el coste computacional y el valor óptimo [60], llegando a la conclusión de que los modelos generados con la base de datos de Intact y Negatome, son más complicados de entrenar y diferenciar de las PPI [61].

3.1 Evaluación de los algoritmos

Tras la elección de los parámetros mas adecuados para cada uno de los modelos que se han llevado a cabo, se procede a entrenarlos con el 75% de los datos que se tiene en cada una de las bases de datos respectivamente [51].

Una vez llevado a cabo el entrenamiento, ya se tienen los modelos finales. Sin embargo, todavía hay que usar el 25% de los datos restantes que no se habían empleado para entrenar cada uno de los modelos, para evaluar la calidad del entrenamiento y así poder comparar que modelo se ajusta mejor [51].

Tras evaluarlo, se comparan los distintos elementos que ha predicho el modelo, con los resultados auténticos de la base de datos en concreto para obtener una tabla como la Tabla 3. En esta tabla se pueden observar los valores acertados y equivocados, predichos por el modelo; y en caso de equivocarse, saber el tipo de fallo cometido.

Tabla 3: Representación en forma de tabla de como quedan los valores tras comparar las predicciones del modelo entrenado y los valores reales de cada uno de los valores estudiados. Fuente: https://blog.aimultiple.com

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Estos valores obtenidos de comparar los valores predichos por el modelo y los valores reales, permiten generar una serie de parámetros que se muestran en las Tablas 4 y 5, además de las Figuras 7 y 8, correspondientes a los algoritmos SVM y RF respectivamente. Estos parámetros calculados ofrecen valores que permiten comparar los modelos de una forma más objetiva.

Tabla 4: Resultados de la exactitud, precisión, sensibilidad, especificidad, coeficiente de correlación de Matthews y la puntuación F1 para los distintos modelos creados con SVM

Database	Features	Accuracy	Precision	Sensitivity	Specifity	MCC	F1 score
Uniprot	AAC	82,17%	92,07%	82,95%	80,00%	58,68%	87,27%
Intact/Negatome	ACC	67,23%	92,33%	68,89%	53,85%	15,08%	78,91%
Uniprot	DPC	82,34%	92,07%	83,14%	80,13%	59,10%	87,38%
Intact/Negatome	DPC	67,74%	94,12%	68,79%	57,41%	16,00%	79,48%
Uniprot	CTD	72,16%	84,14%	76,33%	60,76%	34,79%	80,05%
Intact/Negatome	CTD	67,91%	99,49%	67,53%	84,62%	16,22%	80,46%
Uniprot	PAAC	78,61%	90,03%	80,18%	74,00%	49,97%	84,82%
Intact/Negatome	PAAC	68,59%	10,10%	74,07%	68,33%	18,77%	17,78%

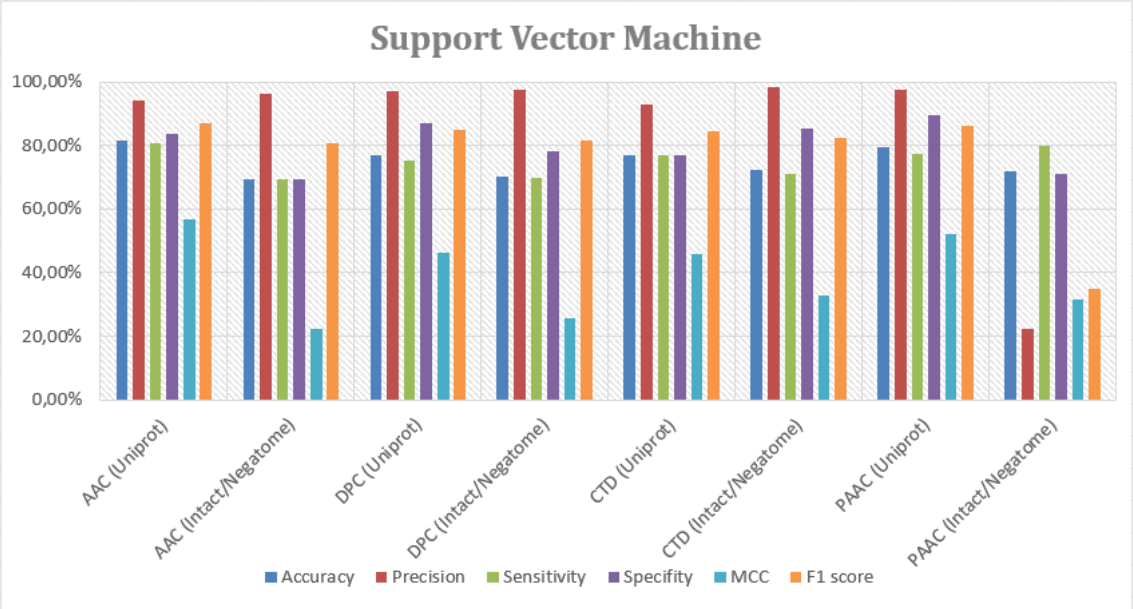

Figura 7: Gráfica donde se reflejan las puntuaciones de los distintos parámetros de cada uno de los modelos geneardos con el algoritmo SVM

En la Tabla 4 se observan los valores obtenidos para los modelos generados con SVM. En esta tabla, el valor de accuracy se observa que es mayor para aquellos modelos que han sido entrenados con nPPIs procedentes de Uniprot, llegando incluso a valores de 83,34% o 82,17% para el modelo cuyas características proteicas se han obtenido con AAC y DPC respectivamente. Además, se observa que estos valores disminuyen cuando la metodología para la obtención de características aumenta su complejidad en los casos estudiados en la base de datos de Uniprot, sin embargo, esto no ocurre cuando se trabaja con Intact y Negatome.

Se puede observar que existe una mayor homogeneidad en los valores de precisión, donde casi todos los valores son superiores al 90%, llegando incluso a valores de 99,49% para el modelo generado a partir de las características CTD y la base de datos nPPI de Intact y negatome. Por otra parte, la precisión del modelo generado con la metodología PAAC y la base de datos de Intact y Negatome es muy baja, teniendo un valor del 10,10%.

Igual que se comentaba en los valores de accuracy, en este caso también se observa que los valores de sensibilidad y especificidad son mayores para los modelos entrenados con nPPIs obtenidas de la base de datos Uniprot respecto a los entrenados con nPPIs obtenidas de la unión de las bases de datos de Intact y Negatome. Aunque existe una excepción para el modelo cuya metodología de obtención de características es CTD, donde la especificidad es menor para Uniprot (60,76%), respecto a Itact y Negatome (84,62%). Aunque en este caso no existe una diferencia destacable entre las medidas de AAC y DPC respecto a CTD y PAAC en la sensibilidad, si se podría encontrar en la especificidad

De una forma similar a la distribución de los valores que se veía reflejada en la precisión, se puede ver que en el parámetro F1 todos los valores están alrededor del 85%. Además, se observa nuevamente que los valores cuyos modelos han sido entrenados con Uniprot, son ligeramente superiores a los de Intact y Negatome. Como ya ocurría anteriormente, el modelo entrenado con nPPIs de Intact y Negatome y cuya metodología empleada para obtener características es PAAC, muestra una puntuación muy baja (17,78%) en comparación al resto de las metodologías

Sin embargo, guiarse únicamente por algunos de estos valores como el valor de accuracy y el valor F1 puede ser un error, ya que tal vez no termina de ajustarse a la realidad, por lo que un buen indicador para terminar de evaluar el modelo, es el parámetro MCC [62]. En este caso, este parámetro refuerza lo que se ha ido observando hasta ahora, ya que la puntuación obtenida por los modelos entrenados con las bases de datos de Intact y Negatome como fuente de nPPIs tienen un valor de MCC cercano a 0, valor que se asemeja a un proceso aleatorio [62], [63].

De forma mas global se puede extraer que los modelos que emplean para obtener las nPPIs la base de datos de Uniprot, proporcionan parámetros con valores mas altos, en la mayor parte de los casos, que los modelos que emplean la base de datos de Intact y Negatome, para aquellos modelos que se han entrenado con SVM. Por lo tanto, cuando se trabaja con SVM, resulta mas eficiente usar como nPPIs proteínas unidas de forma aleatoria [12],[64]; que emplear la base de datos generada con nPPIs procedentes de Intact y Negatome [19].

Por otra parte, cuando la evaluación se centra mas en la metodología de extracción de las características a partir de la estructura primaria, se obtienen mejores resultados en aquellos modelos que se emplean ACC y DPC, que a priori son técnicas mas sencillas que CTD y PAAC. Estos resultados, no se ven reflejados en otros trabajos como [13], [36] donde los resultados no terminan de reflejar diferencias entre las distintas metodologías estudiadas.

Esta diferencia se puede asociar a la combinación de las distintas metodologías con las bases de datos de procedencia de las nPPI. De esta forma, la mejor combinación cuando se emplea el algoritmo SVM es la base de datos Uniprot para obtener las nPPIs junto con AAC y DPC para generar las características de las distintas proteínas. Por otra parte, el peor de los modelos se obtendría entrenándolo con la metodología PAAC y la base de datos de Intact y Negatome como aporte de nPPIs.

Tabla 5: Resultados de la exactitud, precisión, sensibilidad, especificidad, coeficiente de correlación de Matthews y la puntuación F1 para los distintos modelos creados con RF

Database	Features	Accuracy	Precision	Sensitivity	Specifity	MCC	F1 score
Uniprot	AAC	81,49%	94,37%	80,92%	83,46%	56,98%	87,13%
Intact/Negatome	AAC	69,44%	96,42%	69,43%	69,57%	22,15%	80,73%
Uniprot	DPC	77,08%	97,19%	75,40%	87,06%	46,46%	84,92%
Intact/Negatome	DPC	70,29%	97,70%	69,71%	78,05%	25,73%	81,36%
Uniprot	CTD	77,08%	93,09%	77,12%	76,92%	45,64%	84,36%
Intact/Negatome	CTD	72,16%	98,21%	70,98%	85,42%	32,66%	82,40%
Uniprot	PAAC	79,29%	97,44%	77,28%	89,58%	52,28%	86,20%
Intact/Negatome	PAAC	71,99%	22,22%	80,00%	71,16%	31,51%	34,78%

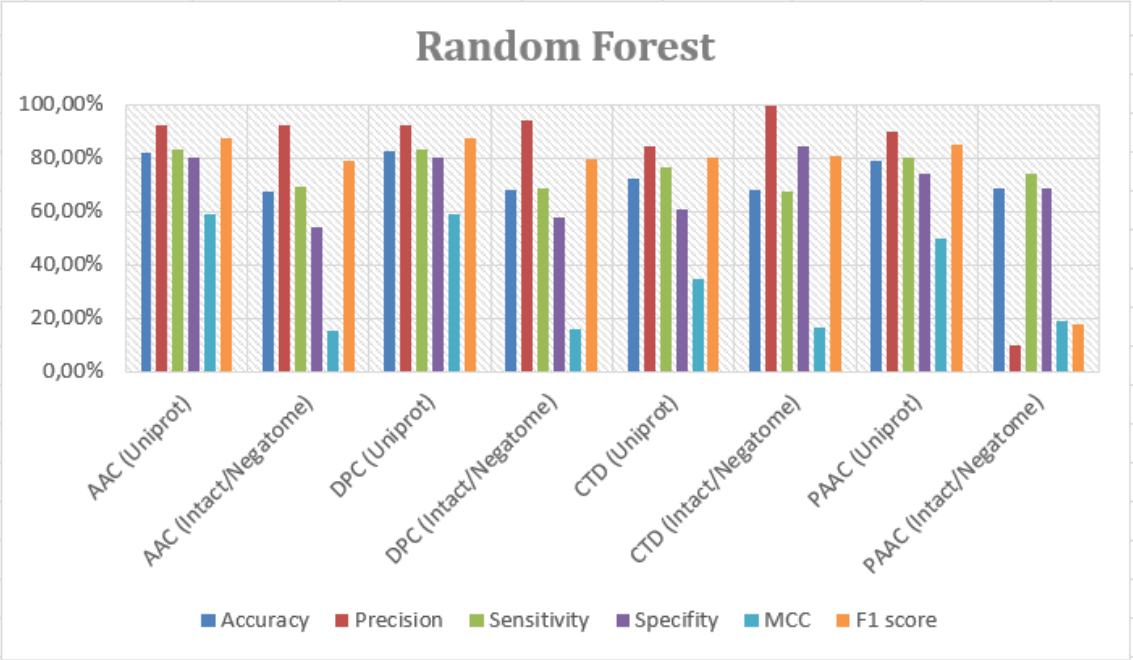

Figura 8: Gráfica donde se reflejan las puntuaciones de los distintos parámetros de cada uno de los modelos geneardos con el algoritmo RF

En la tabla 5, se observan los distintos parámetros que se emplean para evaluar los modelos generados por el algoritmo de RF. Los valores de accuracy obtenidos reflejan valores entre 70% y 80%, siendo el modelo en el que se ha empleado AAC como metodología obtención de características y Uniprot como base de datos de nPPIs el que refleja un mayor valor para este parámetro (81,49%). Además, también se observa que todos los modelos entrenados con la base de datos de Uniprot reflejan valores de mayor tamaño respecto a sus homólogos de Intact y Negatome.

Los datos de precisión son muy elevados, ya que todos los que se observan son superiores al 93%, llegando a valores cercanos a la perfección como el 98,21% que se ha obtenido en el modelo con características obtenidas a partir de CTD y nPPIs procedentes de Intact y Negatome. Esto se cumple en todos los modelos a excepción del modelo entrenado con Intact y Negatome como base de datos de la que se extraen las nPPIs y PAAC como metodología para la obtención de

características a partir de la estructura primaria de las proteínas, donde se ha reflejado un valor del 22,22%.

En el caso de la sensibilidad y especificidad, se producen unos resultados similares a los obtenidos en el caso de los modelos elaborados mediante el algoritmo SVM. Por lo general, los modelos que emplean Uniprot como fuente de nPPIs, reflejan valores mas altos que aquellos que emplean Intact y Negatome; a excepción de la especificidad de los valores cuya metodología empleada para calcular las características de las proteínas es CTD, en la que se invierte esta tendencia. Sin embargo, en este caso, no existe una diferencia tan grande como en SVM entre las técnicas mas sencillas y mas complejas de obtención de características, siendo PAAC la que mayor valor refleja para la especificidad del modelo (89,58% para la base de datos de Uniprot).

En los modelos entrenados empleando el algoritmo RF, se observan unos valores para el parámetro F1 muy homogéneos, alrededor del 85%, para casi todos los modelos. La excepción, al igual que en SVM, se presenta en el modelo que emplea PAAC como metodología de cálculo de características proteicas y la base de datos de Intact y Negatome como fuente de nPPIs. En este modelo, la puntuación generada en el parámetro F1 es de 34,78%, muy inferior al resto.

Igual que en el algoritmo SVM, estudiado anteriormente, el valor MCC, corrobora lo que se estaba viendo hasta el momento con el resto de los parámetros reflejados en la Tabla 5. Aunque la diferencia no sea tan grande como en el anterior, los valores del MCC de los modelos entrenados con Uniprot son mayores que todos aquellos entrenados con Intact y Negatome.

En el caso de los modelos elaborados con RF, los modelos generados con Uniprot como base de datos para la obtención nPPIs obtienen una mejor puntuación en prácticamente todos los parámetros comparándolos con los valores obtenidos en modelos que han usado Intact y Negatome. Sin embargo, la diferencia en este caso no es tan grande como en la Tabla 4 con los modelos generados con SVM, ya que los valores de Intact y Negatome son algo mayores.

Nuevamente, se aprecia que resulta eficiente juntar proteínas de forma aleatoria y tomar estas proteínas como nPPIs para entrenar modelos, lo que hace patente la necesidad de una base de datos de nPPIs de calidad para elaborar el modelo [12], [64].

Respecto a las distintas metodologías para la obtención de características a partir de secuencias proteicas, ocurre lo mismo que en los modelos entrenados con SVM, es decir, las puntuaciones de parámetros son mayores cuando las metodologías son técnicas mas sencillas, sin embargo, en otros trabajos, no se reflejan estas diferencias [39], [65], por lo que estas diferencias se verán influenciadas por la base de datos empleada para la obtención de las nPPIs.

Se podría considerar como el mejor modelo entrenado con RF el modelo que emplea la base de datos Uniprot (es la mejor para todas las metodologías) y la metodología de AAC. Por otra parte, el peor modelo, al igual que en SVM, vuelve a ser el generado con PAAC como metodología para generar características a partir de la estructura primaria y la base de Intact y Negatome para obtener las nPPIs.

Comparando las Tablas 4 y 5, es decir, el algoritmo RF y SVM, a rasgos generales se puede observar que los modelos que mejores resultados aportan en SVM tienen valores ligeramente superiores a los que se generan en RF. Sin embargo, aquellos modelos que empelan las bases de datos de Intact y Negatome, que proporcionan modelos de peor calidad, son mas eficaces cuando se emplea el algoritmo de RF.

4. Conclusiones

4.1 Conclusiones del trabajo

Tras llevar a cabo el estudio y haber analizado los resultados, en este apartado se van a resumir las conclusiones que se han obtenido al realizar este estudio.

- La obtención de nPPIs a partir de la unión aleatoria de proteínas de la base de datos de Uniprot cruzada con SwissProt genera modelos de mayor calidad que aquellos que utilizan las bases de datos de Intact y negatome como aporte de nPPIs.

- Los algoritmos SVM y RF, entrenan mejores modelos de estudio de interacciones proteicas cuando la metodología empleada para la extracción de propiedades a partir de la estructura primaria es mas sencilla, es decir cuando se emplean técnicas como AAC o DPC.

- Los mejores modelos generados con el algoritmo SVM genera presentan puntuaciones ligeramente superiores a los mejores modelos generados con RF.

- Las diferencias entre los modelos que presentan los mejores y los peores parámetros, es menor en cuando se emplea el algoritmo RF que cuando se trabaja con SVM.

4.2 Planificación y metodología

En líneas generales el trabajo se ha realizado conforme a lo planificado. Aunque bien es cierto que no todas las tareas han ocupado el tiempo que se planificó en un principio para su realización, al final aquellas actividades que han costado un periodo mayor del esperado, como la generación de las bases de datos de proteínas; se vieron compensadas con las actividades que se realizaron de forma mas fluida, como el entrenamiento de los algoritmos.

En algunos aspectos, se comenzó con una mentalidad muy ambiciosa, a veces incluso demasiado, que ha llevado a realizar una replanificación del trabajo, como se expone a lo largo de las PEC que se adjuntan como anexos (Anexos 1,2 y 3). Un ejemplo de ello, es el apartado en un principio que prendía realizar un modelo sin machine learning, a partir de alineamiento de secuencias. Sin embargo, debido a limitaciones temporales este análisis no se pudo incluir ya que no se podría llevar a cabo con la calidad que se requiere. Debido a que esta metodología se está quedando algo obsoleta, se prefirió centrar el estudio en los algoritmos de machine learning e invertir el tiempo limitado en poder presentarlos con una mayor calidad.

Por lo demás, creo que el trabajo se ha llevado a cabo en los tiempos estipulados, e incluso se han podido hacer mejoras en la metodología que en un principio no estaban contempladas, como la de generar mas de una base de datos, y emplear mas de una metodología para la obtención de características proteicas, para poder comparar como se ajustan los modelos a distintos datos de partida.

4.3 Perspectivas futuras

Respecto a este trabajo, las perspectivas futuras pasarían principalmente por llevar a cabo un ajuste más exhaustivo de los parámetros de los algoritmos con los que se ha trabajado. Además también se podrían investigar otras de las muchas metodologías que existen para la obtención de características a partir de la estructura primaria de las proteínas como por ejemplo AAIndex [25], [65]. Otro posible campo de mejora sería obtener otra fuente de nPPIs como por ejemplo cambiar las proteínas de Uniprot escogidas aleatoriamente [12] u obtener otras proteínas de interacción para las PPIs.

En este trabajo, se han realizado modelos con 2 de los algoritmos mas empleados en este tipo de estudios, sin embargo, emplear algún otro algoritmo para intentar generar modelos mas robustos, también podría ser una buena forma de mejorar los modelos, como por ejemplo el algoritmo k-nearest neighbours (knn) [14].

Todas estas medidas, podrían contribuir a mejorar la capacidad predictiva de los modelos desarrollados en este trabajo y así poder reducir considerablemente los costes de las medidas experimentales tradicionales que se han empleado hasta ahora y que resultan tan caras y laboriosas [13], [14]

5. Glosario

Ordenadas alfabéticamente:

AAC: Composición de aminoácidos

Aβ 1-42: Péptido amiloide β 1-42

ACC: Exactitud

CTD: Composición/Transición/Distribución

DPC: Composición de dipéptidos

EA: Enfermedad del Alzheimer

EM: Espectrometría de masas

FN: Falso negativo

FP: Falso Positivo

Id: identificador

Knn: k-nearest neighbours

MCC: Coeficiente de correlación de Mathews

nPPI: Proteínas que no interaccionan

PAAC: Composición de pseudo-aminoácidos

PDB: Protein Data Bank

PE: Precisión

PEC: Prueba de evaluación continua

PPI: Proteínas que interaccionan

RBF: Kernel radial

RF: Random Forest

SEN: Sensibilidad

SPE: Especificidad

SVM: Support Vector Machine

TAP: Purificación por afinidad en tándem

TN: Verdaderos negativos

TP: Verdaderos positivos

Y2H: Sistema de 2 híbridos de levaduras

6. Bibliografía

[1] A. Rahman *et al.*, “Sex and Gender Driven Modifiers of Alzheimer’s: The Role for Estrogenic Control Across Age, Race, Medical, and Lifestyle Risks,” *Front. Aging Neurosci.*, vol. 11, no. November, pp. 1–22, 2019.

[2] P.-P. Liu, Y. Xie, X.-Y. Meng, and J.-S. Kang, “History and progress of hypotheses and clinical trials for Alzheimer’s disease,” *Signal Transduct. Target. Ther.*, vol. 4, no. 1, 2019.

[3] H. N. Chan, D. Xu, S. L. Ho, D. He, M. S. Wong, and H. W. Li, “Highly sensitive quantification of alzheimer’s disease biomarkers by aptamer-assisted amplification,” *Theranostics*, vol. 9, no. 10, pp. 2939–2949, 2019.

[4] N. J. Cairns *et al.*, “Bateman et al Biomarker Changes AD 2012,” vol. 367, no. 9, pp. 795–804, 2013.

[5] E. M. Reiman *et al.*, “Brain abnormalities in young adults at genetic risk for autosomal dominant AD,” *Lancet Neurol*, vol. 11, no. 12, pp. 1048–1056, 2012.

[6] A. Association, “2017 Alzheimer’s disease facts and figures,” *Alzheimer’s Dement.*, vol. 13, no. 4, pp. 325–373, 2017.

[7] J. Zissimopoulos, E. Crimmins, and P. St.clair, “The value of delaying alzheimer’s disease onset,” *Forum Heal. Econ. Policy*, vol. 18, no. 1, pp. 25–39, 2015.

[8] R. J. Nicolas W. Cortes-Penfield, Barbara W. Trautner, “乳鼠心肌提取 HHS Public Access,” *Physiol. Behav.*, vol. 176, no. 5, pp. 139–148, 2017.

[9] H. Hampel *et al.*, “Core biological marker candidates of Alzheimer’s disease - Perspectives for diagnosis, prediction of outcome and reflection of biological activity,” *J. Neural Transm.*, vol. 111, no. 3, pp. 247–272, 2004.

[10] L. S. Schneider *et al.*, “Disease : an Appraisal From 1984 To 2014,” vol. 275, no. 3, pp. 251–283, 2015.

[11] F. Saldívar-González, F. D. Prieto-Martínez, and J. L. Medina-Franco, “Descubrimiento y desarrollo de fármacos: un enfoque computacional,” *Educ. Quim.*, vol. 28, no. 1, pp. 51–58, 2017.

[12] L. Zhang, G. Yu, M. Guo, and J. Wang, “Predicting protein-protein interactions using high-quality non-interacting pairs,” *BMC Bioinformatics*, vol. 19, no. Suppl 19, 2018.

[13] Y. Tang, C. Wei, K. Sumikoshi, S. Nakamura, T. Terada, and K. Kadota, “Predicting Protein – Protein Interactions Using Sequence Homology and Machine-Learning Methods,” *Res. J. Life Sci. Bioinformatics, Pharm. Chem. Sci.*, vol. 3, no. 1, pp. 1–26, 2017.

[14] Z. H. You, K. C. C. Chan, and P. Hu, “Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest,” *PLoS One*, vol. 10, no. 5, pp. 1–19, 2015.

[15] K. A. Piez, “Primary Structure,” *Biochem. Collagen*, vol. 17, no. 5, pp. 1–44, 1976.

[16] Z. Ding and D. Kihara, “Computational identification of protein-protein interactions in model plant proteomes,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–13, 2019.

[17] J. Piñero *et al.*, “DisGeNET: A discovery platform for the dynamical exploration of human diseases and their genes,” *Database*, vol. 2015, pp. 1–17, 2015.

[18] S. Orchard *et al.*, “The MIntAct project - IntAct as a common curation platform for 11 molecular interaction databases,” *Nucleic Acids Res.*, vol. 42, no. D1, pp. 358–363, 2014.

[19] P. Blohm *et al.*, “Negatome 2.0: A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis,” *Nucleic Acids Res.*, vol. 42, no. D1, pp. 0–5, 2014.

[20] L. Breuza *et al.*, “The UniProtKB guide to the human proteome,” *Database*, vol. 2016, pp. 1–10, 2016.

[21] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy array: A structure for efficient numerical computation,” *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.

[22] W. McKinney, “pandas: a Foundational Python Library for Data Analysis and Statistics,” *Python High Perform. Sci. Comput.*, no. December, pp. 1–9, 2011.

[23] P. J. A. Cock *et al.*, “Biopython: Freely available Python tools for computational molecular biology and bioinformatics,” *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.

[24] Z. Liu *et al.*, “Mathematical models of amino acid panel for assisting diagnosis of children acute leukemia,” *J. Transl. Med.*, vol. 17, no. 1, pp. 1–11, 2019.

[25] Z. Chen *et al.*, “IFeature: A Python package and web server for features extraction and selection from protein and peptide sequences,” *Bioinformatics*, vol. 34, no. 14, pp. 2499–2502, 2018.

[26] W. McClay, *A Magnetoencephalographic/Encephalographic (MEG/EEG) Brain-Computer Interface Driver for Interactive iOS Mobile Videogame Applications Utilizing the Hadoop Ecosystem, MongoDB, and Cassandra NoSQL Databases*, vol. 6, no. 4. 2018.

[27] P. H. Russell, R. L. Johnson, S. Ananthan, B. Harnke, and N. E. Carlson, “A large-scale analysis of bioinformatics code on GitHub,” *PLoS One*, vol. 13, no. 10, pp. 1–19, 2018.

[28] J. Shen *et al.*, “Predicting protein-protein interactions based only on sequences information,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 11, pp. 4337–4341, 2007.

[29] “DisGeNET - a database of gene-disease associations.” [Online]. Available: https://www.disgenet.org/browser/0/1/0/C0002395/. [Accessed: 10-Oct-2019].

[30] “Índice de /pub/databases/intact/current/psimitab.” [Online]. Available: ftp://ftp.ebi.ac.uk/pub/databases/intact/current/psimitab. [Accessed: 22-Oct-2019].

[31] “The Negatome Database.” [Online]. Available: http://mips.helmholtz-muenchen.de/proj/ppi/negatome1/. [Accessed: 28-Oct-2019].

[32] A. Ben-Hur and W. S. Noble, “Choosing negative examples for the prediction of protein-protein interactions,” *BMC Bioinformatics*, vol. 7, no. SUPPL.1, pp. 1–6, 2006.

[33] “UniProt.” [Online]. Available: https://www.uniprot.org/. [Accessed: 01-Nov-2019].

[34] K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, “GenBank,” *Nucleic Acids Res.*, vol. 44, no. D1, pp. D67–D72, 2016.

[35] P. K. Meher *et al.*, “nifPred: Proteome-wide identification and categorization of nitrogen-fixation proteins of diaztrophs based on composition-transition-distribution features using support vector machine,” *Front. Microbiol.*, vol. 9, no. MAY, pp. 1–16, 2018.

[36] P. K. Meher, T. K. Sahu, A. Banchariya, and A. R. Rao, “DIRProt: A computational approach for discriminating insecticide resistant proteins from non-resistant proteins,” *BMC Bioinformatics*, vol. 18, no. 1, pp. 1–14, 2017.

[37] B. Manavalan, T. H. Shin, and G. Lee, “PVP-SVM: Sequence-based prediction of phage virion proteins using a support vector machine,” *Front. Microbiol.*, vol. 9, no. MAR, pp. 1–10, 2018.

[38] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim, “Prediction of protein folding class using global description of amino acid sequence,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 92, no. 19, pp. 8700–8704, 1995.

[39] B. Manavalan, T. H. Shin, M. O. Kim, and G. Lee, “PIP-EL: A New Ensemble Learning Method for Improved Proinflammatory Peptide Predictions,” *Front. Immunol.*, vol. 9, no. July, p. 1783, 2018.

[40] K. C. Chou, “Prediction of protein cellular attributes using pseudo-amino

acid composition," *Proteins Struct. Funct. Genet.*, vol. 43, no. 3, pp. 246–255, 2001.

[41] P. Du and Y. Yu, "SubMito-PSPCP: Predicting protein submitochondrial locations by hybridizing positional specific physicochemical properties with pseudoamino acid compositions," *Biomed Res. Int.*, vol. 2013, 2013.

[42] J. Liu, Z. Qu, M. Yang, J. Sun, S. Su, and L. Zhang, "Jointly integrating VCF-based variants and OWL- based biomedical ontologies in MongoDB," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. PP, no. c, p. 1, 2019.

[43] "Welcome to Python.org." [Online]. Available: https://www.python.org/. [Accessed: 01-Oct-2019].

[44] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, "Scikit-learn," *GetMobile Mob. Comput. Commun.*, vol. 19, no. 1, pp. 29–33, 2015.

[45] K. M. Mendez, L. Pritchard, S. N. Reinke, and D. I. Broadhurst, "Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing," *Metabolomics*, vol. 15, no. 10, pp. 1–16, 2019.

[46] "pandas · PyPI." [Online]. Available: https://pypi.org/project/pandas/. [Accessed: 01-Dec-2019].

[47] "Biopython · Biopython." [Online]. Available: https://biopython.org/. [Accessed: 19-Dec-2019].

[48] "scikit-learn: machine learning in Python — scikit-learn 0.22 documentation." [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 19-Dec-2019].

[49] "PyMongo 3.9.0 Documentation — PyMongo 3.9.0 documentation." [Online]. Available: https://api.mongodb.com/python/current/. [Accessed: 01-Oct-2019].

[50] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics and Proteomics*, vol. 15, no. 1, pp. 41–51, 2018.

[51] B. Lantz, *Machine Learning with R: Second Edition*. 2015.

[52] "SVM and Kernel SVM - Stupid Simple AI Series - Medium." [Online]. Available: https://medium.com/stupid-simple-ai-series/svm-and-kernel-svm-fed02bef1200. [Accessed: 24-Dec-2019].

[53] "sklearn.model_selection.GridSearchCV — scikit-learn 0.22 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. [Accessed: 01-Dec-2019].

[54] “3.1. Cross-validation: evaluating estimator performance — scikit-learn 0.22 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Accessed: 01-Dec-2019].

[55] “Árboles de decisión: una forma sencilla de visualizar una decisión.” [Online]. Available: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb. [Accessed: 24-Dec-2019].

[56] “Random Forest Algorithm for Machine Learning - Capital One Tech - Medium.” [Online]. Available: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb. [Accessed: 24-Dec-2019].

[57] Z. Qiu, B. Zhou, and J. Yuan, “Protein–protein interaction site predictions with minimum covariance determinant and Mahalanobis distance,” *J. Theor. Biol.*, vol. 433, pp. 57–63, 2017.

[58] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schoelkopf, and a Smola, “Support Vector Machine - Reference Manual,” *J. Mach. Learn. Res.*, vol. 7, no. CSD-TR-98-03, pp. 1687–1712, 1998.

[59] “Parámetros SVM de RBF: documentación de scikit-learn 0.22.” [Online]. Available: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. [Accessed: 28-Dec-2019].

[60] S. Janitza and R. Hornung, *On the overestimation of random forest’s out-of-bag error*, vol. 13, no. 8. 2018.

[61] B. A. Goldstein, A. E. Hubbard, A. Cutler, and L. F. Barcellos, “An application of Random Forests to a genome-wide association dataset: Methodological considerations and new findings,” *BMC Genet.*, vol. 11, 2010.

[62] D. Chicco, “Ten quick tips for machine learning in computational biology,” *BioData Min.*, vol. 10, no. 1, pp. 1–17, 2017.

[63] S. Boughorbel, F. Jarray, and M. El-Anbari, “Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric,” *PLoS One*, vol. 12, no. 6, pp. 1–17, 2017.

[64] N. Zaki, S. Lazarova-Molnar, W. El-Hajj, and P. Campbell, “Protein-protein interaction based on pairwise similarity,” *BMC Bioinformatics*, vol. 10, no. June, 2009.

[65] S. L. Weng *et al.*, “Investigation and identification of protein carbonylation sites based on position-specific amino acid composition and physicochemical features,” *BMC Bioinformatics*, vol. 18, no. Suppl 3, 2017.

7. Anexos

- PEC1
- PEC2
- PEC3
- Scripts en Python y ficheros de partida para el trabajo en el repositorio Github: https://github.com/cperezlopez/TFM-bioinform-tica