RESEARCH

Explorando la Demencia Frontotemporal mediante Biología de Sistemas: Un Enfoque Integrado

Mario Pascual González*, Ainhoa Nerea Santana Bastante, Carmen Rodríguez González, Gonzalo Mesas Aranda and Ainhoa Pérez González

*Correspondence:
pascualgonzalez.mario@uma.es
ETSI Informática, Universidad de Málaga, Málaga, España
Full list of author information is available at the end of the article

Abstract

La demencia frontotemporal es un fenotipo clínico relacionado con la degeneración progresiva de la capacidades cognitivas presente en enfermedades neurodegenerativas. En este trabajo se construido una red de interacción de proteínas a partir de los genes anotados para el fenotipo en la ontología HPO. A partir de esta red se han aplicado tres diferentes algoritmos de clustering, Leiden, Louvain y Fast Gready. Para medir el rendimiento de estos algoritmos y optimizar los parámetros de los algoritmos se ha utilizado tanto la modularidad como una métrica cuantificar el sentido biológico del cluters formados. A partir de los clusters resultantes se ha realizado un análisis de enriquecimiento con el objetivo de identificar los procesos claves relacionados con la demencia frontotemporal.

Keywords: HPO; Demencia Frontotemporal; Biología de Sistemas

1 Introducción

La demencia se define como un síndrome caracterizado por un deterioro cognitivo que produce alteraciones en la memoria, el pensamiento y el comportamiento de una persona. Esto dificulta la capacidad del paciente para realizar sus actividades sociales o laborales. [**?**] Se estima que hay alrededor de 44 millones de personas con demencia, se prevé que esta cifra será más del triple para 2050. [**?**]. La *Enfermedad de Alzheimer* (AD), es la enfermedad más común donde se presenta este síndrome (45-55%), seguida de la demencia vascular y la demencia por cuerpos de Lewy. La demencia frontotemporal no supera el 5% en las frecuencias relativas. [**?**, **?**]. En este proyecto se estudiará un fenotipo concreto presente en varios casos de demencia, la denominada *Demencia Frontotemporal* (FTD).

La FTD es un fenotipo clínico caracterizado por la degeneración progresiva de funciones cognitivas relacionadas con el comportamiento, la personalidad, y el lenguaje, resultado de procesos neurodegenerativos que afectan principalmente a los lóbulos frontal y temporal del cerebro [**?**, **?**, **?**]. Es una de las principales causas de demencia en personas menores de 65 años, presentándose con mayor frecuencia entre los 45 y 65 años [**?**, **?**]. Las variantes clínicas más comunes de la FTD son la variante conductual (bvFTD), experimentada por el 70% de los pacientes y que afecta la regulación del comportamiento y las emociones [**?**, **?**], y la afasia progresiva primaria (PPA), que afecta las habilidades del lenguaje y se subdivide en variantes no fluente/agramática (nfvPPA), semántica (svPPA) y logopénica (lvPPA) [**?**].

La FTD, al tratarse de un conjunto heterogéneo de fenotipos, muestra conexiones significativas con otras enfermedades neurodegenerativas. En estudios de coocurrencia de términos, *Esclerosis Lateral Amiotrófica* (ELA) y EA son conceptos frecuentemente asociados con FTD, lo que sugiere una relación estrecha [**?**]. Se puede deducir entonces otras enfermedades neurodegenerativas presentan el fenotipo clínico de la FTD, indicando que hay una neuropatología subyacente que las relaciona, manifestada bajo este conjunto de fenotipos. La degeneración lobar frontotemporal (FTLD) es el término general que agrupa a los fenotipos patológicos que dan lugar al fenotipo clínico de la FTD [**?**]. La FTLD se clasifica según la acumulación de proteínas anormales en las neuronas. Los subtipos principales incluyen: FTLD-tau, caracterizado por la acumulación de tau hiperfosforilada y asociado fuertemente con la bvFTD [**?**]; FTLD-TDP, que afecta a más del 50% de los pacientes tau-negativos, estrechamente relacionada con la bvFTD y la svPPA [**?**]; y FTLD-FUS [**?**]. El ejemplo más sonado en la literatura se relaciona con la ELA, una forma común de enfermedad de la motoneurona (MND) [**?**], la cual es una enfermedad neurodegenerativa que comparte causas genéticas y neuropatologías con la FTD [**?**]. Mutaciones en genes como *C9orf72* [**?**], *TARDBP* [**?**] y *OPTN* [**?**] se han identificado en pacientes que presentan el fenotipo FTD, padecen ELA, o con manifestaciones de ambas. Las expansiones en *C9orf72* una causa frecuente en ambos casos [**?**, **?**]. Patológicamente, se han observado disfunciones en el sistema autofagia-lisosoma [**?**] e inclusiones citoplásmicas neuronales tau-negativas pero ubiquitina-positivas [**?**] en ambas enfermedades. Pacientes con MND pueden desarrollar afectación cognitiva y evolucionar a FTD [**?**], e incluso mostrar síntomas típicos de bvFTD [**?**]. Asimismo, el parkinsonismo, especialmente el síndrome corticobasal, presenta solapamientos considerables con la FTD, incluyendo trastornos motores y cognitivos [**?**]

En cuanto al diagnóstico, la neuroimagen permite distinguir de manera confiable los subtipos de FTLD de otras demencias, ayudando a correlacionar los síntomas neuropsiquiátricos con los patrones de atrofia cerebral. Técnicas como la Image por Resonancia Magnética (MRI) permiten detectar la atrofia focal en los lóbulos frontal y temporal, típicamente observada en pacientes con bvFTD [**?**]. Las técnicas de neuroimagen funcional, como la tomografía por emisión de positrones (FDG-PET) y la tomografía por emisión de fotón único (SPECT), se utilizan para identificar áreas de hipometabolismo cerebral, dado que bvFTD muestra hipometabolismo en las regiones frontales [**?**, **?**] (Figura 1). Estos métodos también son efectivos para caracterizar las variantes de la PPA. La nfvPPA muestra atrofia en la región fronto-insular, mientras que la svPPA afecta a los lóbulos temporales anteriores [**?**] (Figura 1). Técnicas avanzadas como la PET con amiloide-β, han mostrado ser prometedoras para discriminar entre casos atípicos de AD y FTLD [**?**]. Además, se ha avanzado en la comprensión de las bases genéticas de la FTD, con mutaciones en genes como MAPT, GRN y C9orf72, que están implicadas en aproximadamente el 30-50% de los casos familiares de FTD [**?**].

Siguiendo en esta línea, en torno al 40% de los casos en los que se expresa la FTD son familiares, es decir tienen un patrón hereditario [**?**, **?**]. De los genes que más frecuentemente se ven implicados, uno es el MAPT, que está implicado en la producción de la proteína tau, un importante mediador en la polimerización y estabilización de los microtúbulos cerebrales [**?**]. Algunas mutaciones en el gen GRN

provocan una producción reducida de progranulina, que causa neurodegeneración y se asocia típicamente con bvFTD, aunque también se han reportado casos de PPA [**?**]. La causa genética más común de la FTD, viene dada por el gen C9ORF72, el cual sufre normalmente una expansión del hexanucleótido GGGGCC en una región no codificante del cromosoma 9 [**?**].

Otros genes implicados son el gen TARDBP, del cual se han encontrado mutaciones tanto en pacientes con FTD esporádico como familiar [**?**]; el gen VCP, que interviene en diversos procesos celulares, del cual se han encontrado mutaciones en pacientes con FTD negativos en MAPT, GRN y C9ORF72 [**?**]; o el gen CHMP2B, que sufre mutaciones de truncamiento o sin sentido, relacionado con la demencia frontotemporal del cromosoma 3 (FTD-3) [**?**].

En relación con los factores ambientales que afectan al fenotipo, un estudio sugiere una relación entre el trauma craneoencefálico y la FTD [**?**, **?**]. Además, no se encontró relación con factores como el tabaco, el alcohol, exposición a químicos, pesticidas o insecticidas [**?**].

Cabe destacar que la esperanza de vida promedio de un paciente que presenta los fenotipos clínicos de la FTD es de 6 a 8 años, variando entre 3 y 20 años según la gravedad y las mutaciones genéticas presentes [**?**]. Aunque no existen tratamientos aprobados por la FDA, las estrategias terapéuticas actuales se adaptan al fenotipo predominante en cada paciente, combinando intervenciones farmacológicas y no farmacológicas para mejorar la calidad de vida y controlar los síntomas [**?**].

Entre los tratamientos farmacológicos más prometedores destacan las terapias génicas, como PBFT02, que corrige mutaciones en el GRN [**?**], y otros medicamentos experimentales dirigidos a pacientes con mutaciones en granulina, como el FRM-0334 [**?**]. También se emplean estimulantes del sistema nervioso central y antipsicóticos atípicos para manejar los síntomas conductuales [**?**]. En cuanto a las terapias no farmacológicas, se incluyen intervenciones en el estilo de vida y terapias ocupacionales, del habla y físicas para mejorar la funcionalidad y la comunicación de los pacientes.

2 Objetivos

Se plantean los siguientes objetivos para este estudio:

1. **RQ1**. ¿Es posible identificar los procesos biológicos implicados en la Demencia Frontotemporal mediante el análisis de la red de interacción proteína-proteína asociado?.
2. **RQ2**. ¿Es posible implementar una detección de comunidades sobre la red de interacción proteína-proteína tal que detecte las interacciones clave del fenotipo?
3. **SRQ1**. Bajo el marco de la biología de sistemas, integrar el conocimiento estadístico de la red, los resultados de la detección de comunidades, y el análisis funcional para cumplir con los objetivos expuestos.

3 Materiales y métodos

3.1 Datos

Datos del fenotipo

Para este estudio, se utilizó el fenotipo Demencia frontotemporal, identificado con el término HP:0002145 en Human Phenotype Ontology (HPO). A partir de este

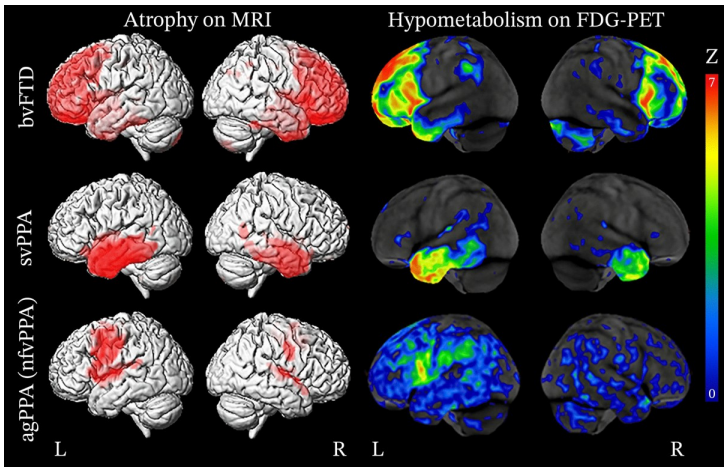

Figure 1: Comparación entre los patrones de atrofia cerebral y el hipometabolismo en diferentes variantes de FTD. En la columna izquierda, reconstrucciones 3D de MRI, que muestran las áreas de atrofia cerebral (en rojo) en tres subtipos de FTD: bvFTD, svPPA, y nfvPPA. En la columna derecha, reconstrucciones 3D de FDG-PET, las cuales indican el hipometabolismo cerebral (en colores) en las mismas regiones afectadas, reflejando la disminución del metabolismo en las áreas afectadas. Los colores cálidos (rojo/amarillo) indican mayor hipometabolismo, mientras que los colores fríos (verde/azul) indican menor hipometabolismo. Imagen tomada de [**?**].

término HPO, se han extraído 52 genes asociados al fenotipo. Estos genes se obtuvieron mediante la API Ontology Annotation Network [**?**] de HPO, que permite acceder programáticamente a las anotaciones entre términos fenotípicos y genes. A través de esta API, se descargaron los datos en formato JSON, que luego fueron procesados para extraer los nombres de los genes y guardarlos en un archivo TSV.

Estos genes, relacionados con el desarrollo de la demencia frontotemporal y otras patologías neurodegenerativas, representan el conjunto inicial de genes sobre el que se construirá la red de interacciones para el análisis posterior. Cada uno de ellos se identifica mediante su ID único en la base de datos de NCBIGene, lo cual facilita el acceso y la referencia a los datos genéticos específicos.

Para asegurar la reproducibilidad, se utilizó la versión 2.0.4 de HPO [**?**], para obtener el término fenotípico y descargar los genes relacionados con el fenotipo de estudio. En la siguiente sección se proporciona una descripción detallada de HPO.

Human Phenotype Ontology (HPO)

HPO proporciona una ontología estandarizada que describe anomalías fenotípicas observadas en enfermedades humanas, facilitando la identificación y análisis de genes asociados a diversas características clínicas. Cada término en HPO representa una anomalía específica, como la demencia frontotemporal, y está diseñado para facilitar la caracterización precisa de los fenotipos en el contexto de enfermedades hereditarias. La ontología se desarrolla y actualiza de forma continua utilizando fuentes como la literatura médica, así como bases de datos como Orphanet, DECI-

PHER y OMIM. Actualmente, HPO contiene más de 18,000 términos y ofrece más de 156,000 anotaciones asociadas a enfermedades hereditarias [**?**].

Datos de interacción

Los datos de interacción representan conexiones funcionales y físicas entre proteínas, y constituyen la base para construir redes de interacción en el análisis de procesos biológicos. En este estudio, los datos de interacción proteína-proteína (PPI) fueron extraídos de la base de datos STRING mediante su API REST [**?**], que permite recuperar programáticamente redes de interacción específicas basadas en listas de genes o proteínas de interés.

A través de esta API, se obtuvieron las interacciones entre los genes asociados al fenotipo FTD (HP:0002145) en formato TSV. En este archivo, cada fila representa una interacción entre dos proteínas y contiene las siguientes columnas.

- **protein1:** ID de la primera proteína en la interacción, precedido por el código taxonómico del organismo (por ejemplo, "9606" para proteínas humanas).
- **protein2:** ID de la segunda proteína en la interacción, también con el prefijo de organismo.
- **combined_score:** Puntuación de confianza combinada para cada interacción proteína-proteína, con valores que oscilan entre 0 y 1000. Esta puntuación refleja la probabilidad de que una interacción sea real, basada en una integración de diversas fuentes de evidencia, como co-ocurrencia filogenética, co-expresión, minería de texto y datos experimentales. Cada tipo de evidencia se evalúa y puntúa individualmente, y luego se combina en el "combined_score", proporcionando así un indicador global de confiabilidad para cada interacción funcional o física [**?**].

Estos datos obtenidos se utilizarán para construir una red de interacciones entre los genes asociados al fenotipo FTD, permitiendo analizar las relaciones funcionales entre proteínas en este contexto. Esta red servirá como base para el análisis de clustering, facilitando la identificación de módulos de genes potencialmente implicados en funciones biológicas específicas.

Para asegurar la reproducibilidad del análisis, se utilizó la versión 12.0 de STRING, junto con su API REST de STRING para la extracción de interacciones.

STRING

La base de datos STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) es un recurso bioinformático diseñado para recopilar, organizar y analizar redes de interacciones proteína-proteína y asociaciones funcionales en cualquier genoma secuenciado. STRING integra información de diversas fuentes, como minería de texto científico, predicciones computacionales basadas en coexpresión y contexto genómico, y datos experimentales obtenidos de estudios de interacciones proteicas. Además, los usuarios pueden acceder a la base de datos para explorar redes de interacción, realizar análisis de enriquecimiento funcional y generar redes personalizadas para genomas específicos, facilitando así la investigación en biología celular y molecular. Actualmente, STRING cubre 59.309.604 proteínas provenientes de 12.535 organismos. [**?**].

3.2 Software

Para el análisis funcional y la construcción de redes genéticas en este estudio, se seleccionaron herramientas especializadas que permiten tanto la exploración bioinformática como la visualización de datos complejos. Dado que el objetivo principal es investigar la interacción entre genes y módulos específicos asociados a la demencia frontotemporal, se ha optado por una combinación de paquetes en Python que ofrecen un balance entre precisión analítica y capacidades visuales avanzadas.

Paquetes de Python para el análisis funcional y otras funciones

- **NumPy (versión 1.26.0)**: Esta librería proporciona soporte para arrays multidimensionales y funciones matemáticas de alto rendimiento. Es fundamental para realizar cálculos numéricos eficientes y operaciones matemáticas en grandes conjuntos de datos [**?**].
- **Pandas (versión 2.2.3)**: Este paquete proporciona estructuras de datos eficientes y flexibles, como DataFrames, que facilitan el procesamiento y manipulación de datos complejos. En el contexto de este estudio, Pandas permite organizar, filtrar y procesar resultados de enriquecimiento funcional, simplificando el manejo de grandes volúmenes de datos bioinformáticos [**?**].
- **Stringdb (versión 0.1.5)**: Esta librería proporciona una interfaz en Python para acceder a la base de datos STRING, recuperando información sobre interacciones proteína-proteína. Además, permite integrar estos datos en análisis de redes y enriquecimiento funcional, facilitando la investigación en biología de sistemas. [**?**].
- **Requests (versión 2.31.0)**: Requests es una biblioteca para realizar solicitudes HTTP de manera simple y efectiva. En este estudio, se utiliza para conectar con APIs externas como la de STRINGdb, permitiendo la descarga automatizada de datos de redes y enriquecimiento [**?**].
- **GSEAPY (versión 1.0.5)**: GSEAPY permite realizar análisis de enriquecimiento funcional y análisis de vías de manera sencilla, integrando métodos como GSEA y prerank para conjuntos de genes. Es fundamental para identificar procesos biológicos relevantes en los genes analizados [**?**].
- **Igraph (versión 0.10.8)**: Una biblioteca para la creación, manipulación y visualización de grafos y redes complejas. Es esencial para representar y analizar redes de interacciones génicas y otros tipos de grafos en el contexto de estudios bioinformáticos [**?**].
- **NetworkX (versión 3.1)**: Esta librería facilita el análisis y visualización de estructuras de grafos y redes complejas. Se aplica para modelar y visualizar redes de interacciones génicas [**?**].
- **Matplotlib (versión 3.8.1)**: Matplotlib es una librería de visualización muy versátil que soporta múltiples tipos de gráficos en 2D, lo que resulta útil para representar tendencias y relaciones entre genes en gráficos de líneas, barras, dispersión, y más. Se pondrán en uso extensiones como las de Colors, CM, Axes, Patches. Este paquete se utilizará para visualizar los resultados de enriquecimiento y las interacciones génicas [**?**].
- **Matplotlib-Venn (versión 0.11.9)**: Esta librería permite crear diagramas de Venn con Matplotlib para representar gráficamente las relaciones entre conjuntos de datos [**?**].

- **UpSetPlot (versión 0.8.0)**: UpSetPlot es una herramienta para crear gráficos de tipo UpSet, que permiten visualizar intersecciones entre conjuntos de datos, útil en el análisis de enriquecimiento funcional [**?**].
- **Seaborn (versión 0.12.2)**: Seaborn es una librería de visualización basada en Matplotlib que permite crear gráficos estadísticos atractivos y con estilo. Se utiliza para explorar y visualizar patrones y relaciones en los datos [**?**].
- **Plotly (versión 5.18.0)**: Plotly es una biblioteca interactiva para crear gráficos y visualizaciones avanzadas. Se utiliza para representar redes e interacciones complejas de forma dinámica y visualmente atractiva [**?**].
- **Scienceplots (versión 2.1.1)**: Este paquete extiende Matplotlib proporcionando estilos de gráficos estéticamente optimizados para publicaciones científicas. Con Scienceplots, se puede lograr una presentación visual de alta calidad, ideal para gráficos que requieren una apariencia profesional [**?**].
- **Argparse (versión 1.4.0)**: Argparse es una librería estándar para analizar argumentos en la línea de comandos, facilitando la ejecución flexible de scripts y automatización de tareas [**?**].
- **Logging (versión 0.5.1.2)**: Logging es un módulo estándar que permite registrar mensajes y eventos durante la ejecución de los scripts, lo que facilita el seguimiento y depuración del proceso de análisis [**?**].
- **Shutil (módulo estándar)**: Permite realizar operaciones con archivos y directorios, como copiar, mover o eliminar, facilitando el manejo del sistema de archivos en los scripts [**?**].
- **JSON (módulo estándar)**: Este módulo facilita la manipulación y el intercambio de datos en formato JSON, útil para procesar configuraciones y respuestas de APIs [**?**].
- **Time (módulo estándar)**: Proporciona funciones para medir el tiempo de ejecución y realizar pausas en los scripts, lo que facilita el control del flujo de trabajo [**?**].
- **Re (módulo estándar)**: Este módulo permite realizar operaciones con expresiones regulares, facilitando la búsqueda y manipulación de cadenas de texto [**?**].
- **Platform (módulo estándar)**: Proporciona información sobre el sistema operativo y el entorno de ejecución, útil para adaptar el comportamiento de los scripts según el entorno [**?**].
- **Random (módulo estándar)**: Facilita la generación de números aleatorios, útil en procesos de muestreo o simulación [**?**].
- **Psutil (versión 5.9.5)**: Permite monitorizar el uso de recursos del sistema, como memoria y CPU, lo cual es útil para optimizar el rendimiento de los scripts [**?**].
- **YAML (versión 6.0)**: Facilita la lectura y escritura de archivos YAML, útil para manejar configuraciones en un formato legible y estructurado [**?**].
- **Optuna (versión 3.5.0)**: Una librería para la optimización automática de hiperparámetros en modelos y algoritmos, mejorando el rendimiento del análisis [**?**].
- **Math (módulo estándar)**: Proporciona funciones matemáticas básicas como logaritmos y raíces cuadradas, necesarias para realizar cálculos en el análisis funcional [**?**].

- **Itertools (módulo estándar)**: Proporciona funciones para crear iteradores eficientes, como combinaciones y permutaciones, útiles para el análisis de redes y conjuntos de datos [**?**].
- **Jinja2 (versión 3.1.3)**: Un motor de plantillas para Python que permite generar contenido dinámico, útil para personalizar informes y representaciones visuales [**?**].
- **Kaleido (versión 0.2.1)**: Una herramienta para exportar gráficos de Plotly a formatos estáticos como PNG, PDF y SVG, facilitando la creación de visualizaciones de alta calidad [**?**].

3.3 Clustering

Al aplicar algoritmos de clustering, nuestro objetivo es descubrir comunidades funcionales dentro de la red de genes asociada a la demencia frontotemporal. Estas comunidades, módulos, o *clusters* funcionales pueden representar procesos biológicos específicos, vías celulares, o mecanismos asociados al fenotipo FTD. Encontrar estos clusters podría revelar posibles dianas terapéuticas o grupos de biomarcadores dentro del conjunto de genes, lo cual podría abrir las puertas a nuevos tratamientos para los pacientes de FTD.

3.3.1 Medidas de Rendimiento

Las medidas de rendimiento usadas para la evaluación de los resultados de clustering usando los algoritmos de la sección anterior, son las siguientes:

- **Modularidad (Q)**: esta medida, definida en la Ecuación 1, es un valor escalar entre -1 y 1 que representa la diferencia entre la densidad de aristas dentro de las comunidades y la densidad de aristas esperada si fuese una red aleatoria con la misma distribución de grados. Esta se define como:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{1}$$

  donde A_{ij} es el peso de la arista entre los nodos i y j, k_i y k_j son las sumas de los pesos de las aristas conectadas a los nodos i y j, c_i y c_j representan las comunidades a las que pertenecen los nodos i y j, y $\delta(c_i, c_j)$ es una función delta de Kronecker que es 1 si $c_i = c_j$ y 0 en caso contrario. El término m es la suma total de los pesos de las aristas en la red.
- **Puntuaje de Enriquecimiento Funcional (PEF)**: esta métrica ha sido ideada con la finalidad de cuantificar el sentido biológico de un clustering. Para ello, se ha ha realizado un enriquecimiento funcional de cada cluster usando la API de StringDB. Para cada cluster, se calcula el PEF local usando estas dos medidas:
  - *P-valor*: de cada término enriquecido, se obtiene el p-valor asociado, que es un valor entre 0 y 1 que mide la confianza que hay en dicho término.
  - *Profundidad*: usando la API de Gene Ontology (GO), se obtiene la profundidad de cada término en la ontología. Este valor se encuentra entre el intervalo [0, 20]. Cuanto más profundo es un término, mayor es normalmente la cantidad de información que ofrece. Esto no siempre se cumple

ya que para un mismo valor de profundidad, dos términos GO puede tener una cantidad de información distinta. Esto es una limitación de la métrica, que podría tenerse en cuenta a la hora de proponer otra versión que ofrezca una mejor aproximación.

Una vez obtenidas estas dos medidas de cada termino enriquecido, se usa la ecuación 2 para obtener el valor de PEF.

$$PEF = \frac{-\log_{10}(\text{p-valor}) \cdot \text{profundidad}}{c}, \tag{2}$$

El objetivo de esta ecuación es combinar de la mejor manera las dos medidas. Por un lado, al p-valor se le calcula el logaritmo en base 10, que equivale al valor del exponente del mismo en notación científica. Además al ser este un valor negativo, se positiviza multiplicándolo por -1. De esta manera, un valor más pequeño del p-valor resultará en un mayor PEF. El logaritmo se usa para estabilizar los valores de la métrica, debido a que los p-valores pueden estar en escalas muy pequeñas, por lo que a veces cambios muy pequeños en este, podría resultar en un gran cambio del PEF. Por último, este valor se multiplica por la profundidad y se divide por una constante c. Esta constante es un supuesto máximo valor teórico para la métrica, que hemos fijado en 600 (-log10(p-valor) = 60, profundidad = 10). Tiene como finalidad ajustar el rango de valores entre 0 y 1, para que se comparable con la modularidad en la optimización multi-objetivo.

Una vez calculado el PEF de un término, se promedian todos los PEFs de cada cluster, y finalmente los de cada cluster para obtener el valor final.

Cuantificar el sentido biológico es una tarea difícil, a la cual pueden haber muchas aproximaciones. Esta propuesta tiene limitaciones, ya que se trata de una primera aproximación a un problema bastante complejo. No obstante, ofrece una aproximación decente y rápida, sirviendo como punto de partida para futuras mejoras.

3.3.2 Algoritmos

A continuación, se detallan los tres algoritmos de clustering, proporcionados por la librería *iGraph*, elegidos para este estudio, los cuales pretenden cubrir diferentes enfoques teóricos en la detección de comunidades funcionales [**?**].

- **Fast Greedy**: es un algoritmo de clustering jerárquico que optimiza directamente la modularidad, lo que le confiere una gran utilidad en biología de sistemas, ya que la esta captura la idea de que los nodos dentro de una comunidad son más conexos entre sí que con nodos de otras comunidades.

  La estrategia que sigue Fast Greedy es voraz, es decir que toma decisiones locales en cada iteración para optimizar la modularidad. Sigue los siguientes pasos: inicialización (considera que cada nodo es un cluster), fusión (en cada iteración fusiona las comunidades que aumenten en mayor medida la modularidad) y terminación (el algoritmo termina cuando no se pueda incrementar más la modularidad) [**?**].

Este algoritmo no precisa del ajuste de ningún hiperparámetro, por lo que los resultados del mismo se han tomado como referencia y punto de partida para los demás algoritmos.

- **Algoritmo Louvain**: este algoritmo es uno de los más utilizados para la detección de comunidades en redes. Al igual que el algoritmo Fast Greedy, se basa en optimizar la modularidad de manera jerárquica. Tiene dos fases claves en su funcionamiento:
  - *Optimización local de modularidad*: al inicio, se asigna a cada nodo su propio cluster. En cada iteración, se evalúa si mover un nodo a la comunidad de uno de sus vecinos incrementa la modularidad de la red. El nodo se mueve a la comunidad que maximiza la modularidad local.
  - *Construcción de la red*: una vez los nodos están en comunidades correspondientes, se agrupan las comunidades en un nuevo "supernodo" y se construye una nueva red en la que los nodos son las comunidades encontradas. Se vuelve a calcular la modularidad y se repite el proceso hasta que no se pueda mejorar más la modularidad.

  Este proceso jerárquico permite detectar comunidades a diferentes escalas de la red [**?**].

  Se ajustó el parámetro de resolución, que controla el tamaño final de las comunidades. El resto de parámetros se dejaron con sus valores por defecto.
- **Algoritmo de Leiden**: este algoritmo se diseñó para mejorar las limitaciones del algoritmo de Louvain, particularmente en términos de garantizar comunidades bien conectadas. Sigue los siguintes pasos:
  - *Movimiento local de nodos*: Cada nodo del grafo comienza en su propia comunidad. El algoritmo evalúa si mover un nodo a la comunidad de uno de sus vecinos incrementa la modularidad. Si es así, el nodo se mueve. Este proceso se repite hasta que ningún movimiento adicional mejore la modularidad.
  - *División interna de comunidades*: Dentro de cada comunidad, el algoritmo verifica si estas son completamente conexas. Si no lo son, divide las comunidades en subcomunidades más pequeñas para garantizar que todas sean subgrafos conexos.
  - *Agregación y simplificación del grafo*: Cada comunidad identificada se trata como un solo nodo, y se construye un nuevo grafo 'resumido'. Luego, se repiten los pasos anteriores con este nuevo grafo [**?**].

  Se ajustó el parámetro γ, así como el número de iteraciones del algoritmo, permitiendo que el Leiden refinara iterativamente la partición. El resto de parámetros se fijaron a su valor por defecto.

3.3.3 Optimización

Como se explicó en la Sección 3.3.2, Louvain y Leiden son algoritmos de clustering con múltiples parámetros configurables. Ajustar adecuadamente estos parámetros es crucial para mejorar tanto la interpretación biológica de la red como la detección de comunidades. Este proceso, conocido como ajuste de hiperparámetros, busca optimizar el rendimiento del algoritmo según diversas métricas. En este apartado, se detalla el procedimiento empleado para ajustar el parámetro de resolución (γ) en

Leiden y Louvain, con el objetivo de maximizar las métricas descritas en la Sección 3.3.1.

La estadística bayesiana es un enfoque probabilístico que utiliza el teorema de Bayes para actualizar las creencias sobre un modelo a medida que se incorporan nuevos datos. El ajuste bayesiano de hiperparámetros (BHO) aplica este enfoque para optimizar los parámetros de un modelo, construyendo y actualizando iterativamente un modelo probabilístico de la función objetivo en función de los hiperparámetros. Este modelo probabilístico sugiere los hiperparámetros a probar en cada iteración, permitiendo enfocar eficientemente la búsqueda en las regiones más prometedoras del espacio de hiperparámetros.

Existen diversos algoritmos basados en BHO, cada uno caracterizado por el modelo probabilístico que construye. En este proyecto, se ha optado por el Tree-structured Parzen Estimator (TPE). En el TPE, se considera que y es el valor obtenido al evaluar la función objetivo utilizando un conjunto de hiperparámetros θ. Se define un umbral y^* que permite dividir nuestros datos en dos conjuntos: los hiperparámetros que resultan en un rendimiento mejor que y^* y los que resultan en un rendimiento peor:

$$\begin{aligned} \mathcal{C}_1 &= \left\{ \theta^{(i)} \mid y^{(i)} \leq y^* \right\} \\ \mathcal{C}_2 &= \left\{ \theta^{(i)} \mid y^{(i)} > y^* \right\} \end{aligned} \tag{3}$$

Usando el método de ventana deslizante de Parzen-Rosenblatt para estimar la función de probabilidad de densidad, se estiman las distribuciones de probabilidad para estos dos conjuntos de hiperparámetros - para más detalles sobre el método de Parzen-Rosenblatt, revisar la Sección 7:

$$\begin{aligned} l(\theta) &= p(\theta \mid \theta \in \mathcal{C}_1) \\ g(\theta) &= p(\theta \mid \theta \in \mathcal{C}_2) \end{aligned} \tag{4}$$

El objetivo es seleccionar nuevos hiperparámetros θ que maximicen el Expected Improvement (EI), que es proporcional al cociente de estas dos densidades:

$$\mathrm{EI}(\theta) \propto \frac{l(\theta)}{g(\theta)} \tag{5}$$

Al maximizar $\mathrm{EI}(\theta)$, se favorecen los hiperparámetros que son más probables en el conjunto de buen rendimiento $\mathcal{C}_1$ y menos probables en el conjunto de peor rendimiento $\mathcal{C}_2$. Este proceso se repite iterativamente; en cada iteración, los nuevos datos de rendimiento actualizan $l(\theta)$ y $g(\theta)$, permitiendo explorar eficientemente el espacio de hiperparámetros.

Para Louvain y Leiden, se han ejecutado, respectivamente, 150 iteraciones, explorando resoluciones entre 0.1 y 2.0 que maximicen tanto el coeficiente Q como el puntuaje FES, descritos en la Sección 3.3.1. Los datos se han almacenado en

una base de datos SQLite, lo que permite extraer informes tabulados y reanudar el ajuste desde el punto de guardado si se desea, facilitando la gestión y continuidad del proceso.

Se generaron varias configuraciones de modelos al ajustar sistemáticamente los parámetros de los algoritmos y evaluamos cada configuración en base a modularidad y puntaje de Enriquecimiento Funcional. Al trazar estos puntajes en un frente de Pareto, identificamos las configuraciones que representan el mejor equilibrio entre coherencia estructural (modularidad) y relevancia biológica (enriquecimiento funcional) [**?**, **?**, **?**]. Las configuraciones a lo largo de esta frontera representan soluciones Pareto-eficientes de clustering, donde mejorar una métrica no compromete significativamente la otra, permitiendo explorar soluciones con un balance diferente de las métricas de rendimiento elegidas.

3.4 Análisis de enriquecimiento de vías biológicas

El análisis de enriquecimiento permite identificar vías biológicas o pathways que están significativamente representados en una lista de genes de interés, por medio de pruebas estadísticas. Un pathway es un conjunto de genes que trabajan en conjunto para llevar a cabo un proceso biológico específico.[**?**]

En este estudio, se realizó un análisis de enriquecimiento para las agrupaciones de genes derivadas del algoritmo de clustering que ha generado un mejor resultado.

Las herramientas de análisis funcional mapean las listas de genes de interés a terminos biólógicos anotados, como los términos GO y utilizan métodos estadísticos, como el test de Ficher o test hipergeométricos, para evaluar el enriquecimiento.[**?**] En particular, el test hipergeométrico evalúa si la representación de cada término funcional en el conjunto de genes de interés es mayor que la esperada por azar.

Para la realización de este análisis funcional se ha hecho uso del módulo *enrich* de biblioteca de Python *GSEAPY* que permite la realización de un análisis de enriquecimiento de una lista de genes mediante la utilización de la API de Enrich. Esta herramienta evalúa si un conjunto de genes de entrada se superpone significativamente con conjuntos de genes previamente anotados. El análisis se ha realiza utilizando la base de datos GO_Biological_Process_2021, que recopila información sobre procesos biológicos. Sin embargo, la utilización de otras ontologías como Gene Ontology (GO), UniProt, Reactome o KEGG podrían información complementaria sobre diferentes tipos de términos funcionales.

Enrich va a proporcionar varias métricasa a cerca de los resultados del enriquecimiento: p-valor, p-valor ajustado, odds ratio y combined score. El p-valor, calculado las pruebas estadísticas anteriormente mencionadas evalúa la significancia estadística de la superposición entre los genes de interés y los anotados en los términos. El p-valor ajustado utilizando el método de Benjamini-Hochberg para controlar la tasa de falsos positivo. La métrica de odds ratio mide la proporción entre los genes superpuestos observados y los esperados bajo una distribución aleatoria. Finalmente, combined score combina el p-value y el odds ratio en una métrica única. [**?**]

Finalmente, se aplicaron filtros para seleccionar únicamente los términos funcionales más relevantes en los resultados obtenidos. Estos filtros han sido el p-valor ajustado a 0.005, combined score a 2000 y el porcentaje de overlap que se ha definico

como el número de genes de la lista de interés que se superponen con los genes anotados del término entre el total de genes anotados en el término a 0.1.

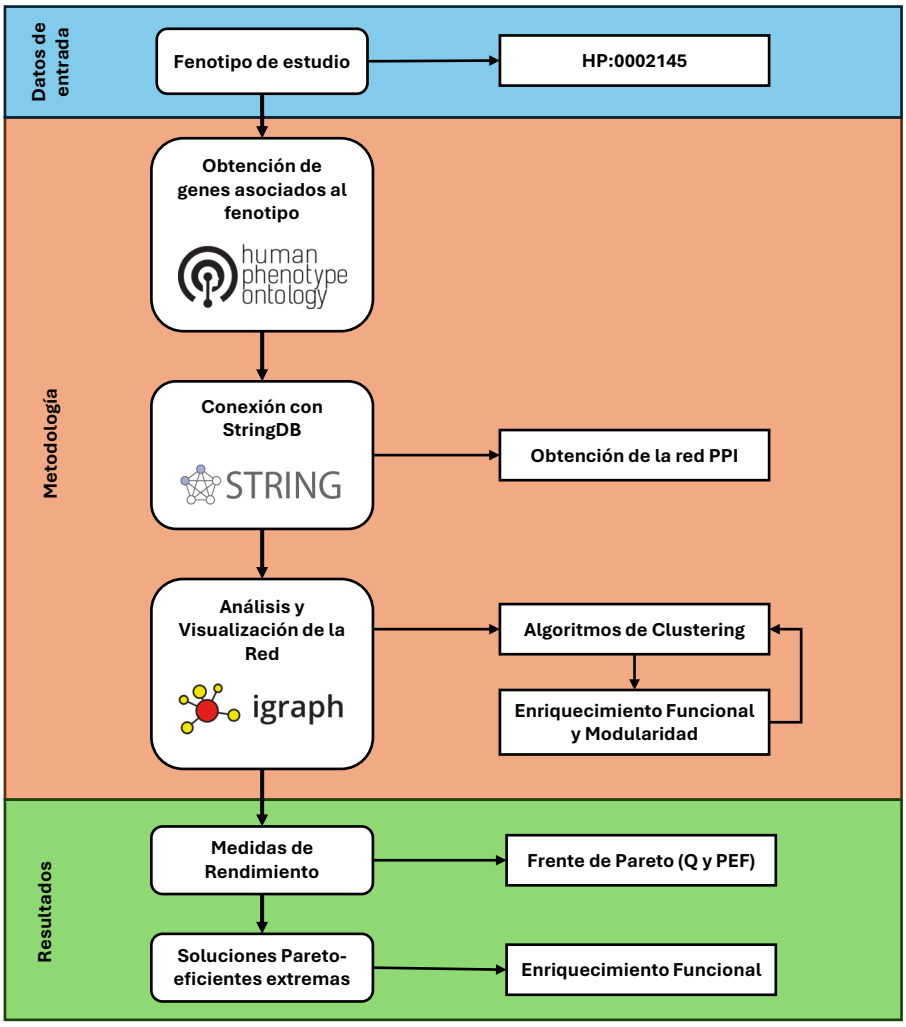

Figure 2: Flujo de trabajo que se seguirá durante el desarrollo del proyecto, partiendo del fenotipo como dato de entrada, metodología y análisis de resultados.

4 Resultados

En esta sección se exponen las visualizaciones y tablas generadas a partir de los resultados de análisis de la red, optimización del clustering, y análisis funcional de las comunidades obtenidas.

4.1 Red PPI y sus propiedades

HPO permite extraer los genes anotados a un fenotipo, por lo que estos han sido descargados y utilizando la API de StringDB se han obtenido la red de interacción (*Figura* 3). Se ha establecido el umbral de *combine_score* a 700.

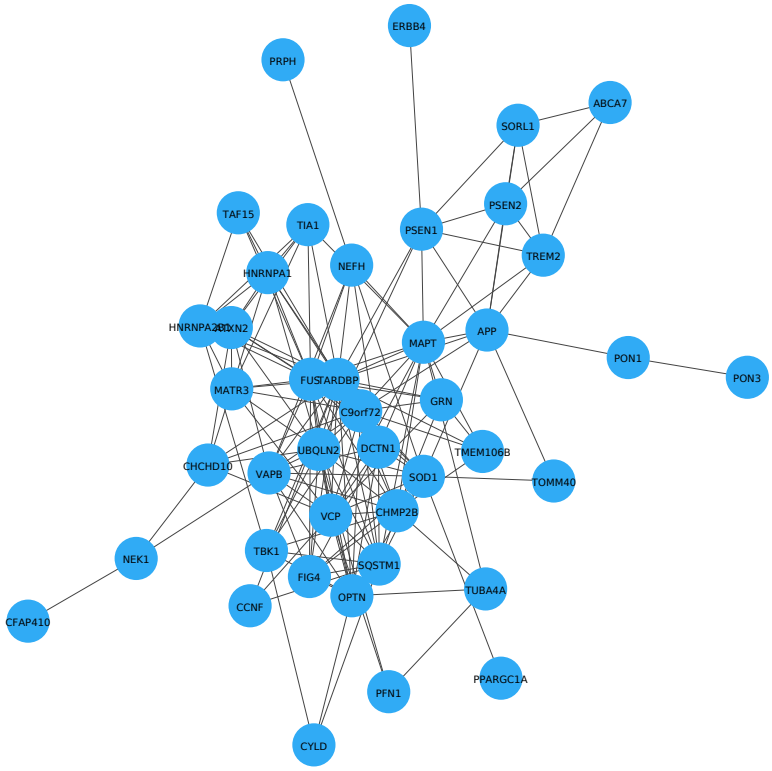

Figure 3: Red de interacción proteína-proteína de los genes del HPO (HP:0002145) obtenida mediante la API de StringDB con un filtro de 400 del combine score

En la *Figura 3* observamos que proteínas con un número de conexiones respecto a otras. Un análisis en mayor profundidad de la red se presenta en la *Tabla 1.*

En HPO se tenían 52 genes relacionados con nuestro fenotipo, pero en la red filtrado tan solo hay 42 proteínas. Por tanto hay diez genes, que no interaccionan con las otras proteínas de la red con un alto umbral de confianza. Con un umbral inferior al establecido, estos genes estarían en la red, pero la significancia biología de las interacciones no sería fiable.

El grado medio de los nodos, correspondiente al número de conexiones de un nodo, es 8.14, un valor considerablemente alto si tenemos en cuenta el número de nodos

Table 1: Resumen de Métricas de la Red PPI

Categoría	Métrica y Valor
Tamaño de la Red	Número de nodos: 42 Número de aristas: 171
Grado	Grado promedio: 8.14 Desviación estándar del grado: 5.93
Conectividad	Grafo conectado: Sí Conectividad de nodos: 1 Conectividad de aristas: 1
Densidad y Sparsity	Densidad: 0.199 Esparcidad: 0.801
Cercanía (Closeness)	Cercanía promedio: 0.456 Desviación estándar de cercanía: 0.093
Centralidad (Betweenness)	Betweenness promedio: 26.50 Desviación estándar de betweenness: 39.56
Transitividad	Transitividad local promedio: 0.550 Desviación estándar de transitividad local: 0.322 Transitividad global: 0.520
Otras Propiedades	Asortatividad: -0.018 Diámetro: 6 Longitud de camino promedio: 2.293

presentes en la red. En total, la red tiene 171 conexiones ente proteínas. Hay un valor significativo de desviación típica del grado de los nodos. Un mejor análisis de las distribución de los grados se presenta en la *Figura 4*. En esta figura, observamos que la gran mayoría de las proteínas, tienen grados relativamente bajos, inferiores a la medio. Encontramos un par de proteínas, con grados superiores a 20, lo que indica que están conectados a más de la mitad de las proteínas de la red. Estos nodos altamente conectados podrían ser hubs (FUS, TARDBP). Si nos fijamos en la densidad del grafo, vemos que tenemos una red con baja densidad, hay pocas conexiones con respecto a las posibles.

La red obtenida es un grafo conexo (ver *Figura 3*), el análisis de la conectivadad llevado a cabo muestra que existe una conectividad tanto de vértice (nodo) como de arista de 1. Esto indica que exiten uno o varios nodos/aristas que si son eliminados desconectan el grafo. Se ha determinado que estos nodos son las proteínas NEK1, SOD1, APP, PON1, NEFH Y PSEN1.

Las siguientes métrica analizadas son la centraliad y cercanía, en la *Tabla 1* se muestra la media y desviación, ya que se tratan de métricas que se miden por cada nodo. Para ambas métricas destaca una de las dos proteínas antes mencionadas con el grado,TARDBP.

Finalmente podemos mencionar la transitividad, probabilidad de que los vecinos de un nodo estén también conectados entre sí. Tanto la transitividad local como la global tienen valores parecidos, entorno a 0.5. Puntualizar que para el calculo de la transitividad local, se han obviado aquellos nodos que dan valores de NaN debido a que solo presentan un vecino.

Figure 4: Distribución de los grados de los nodos, mediante un histograma, de la red de interacción.

4.2 Optimización de Hiperparámetros

El proceso de optimización de hiperparámetros mediante el BHO se analizará mostrando el frente de pareto de ambos algoritmos y una visualización del rendimiento marginal de ambas métricas en base al valor de resolución (γ) evaluado.

4.3 Clustering

Se presentan los resultados de los algoritmos de clustering. Se tiene el baseline, Fast-Greedy, y las soluciones extremas Pareto-Óptimas de Leiden y Louvain.

4.4 Análisis Funcional

Se ha realizado un análisis funcional inicial que ha identificado más de 1200 términos GO asociados con el fenotipo de demencia frontotemporal (DFT), utilizando el algoritmo Leiden y optimizando la modularidad máxima. Tras aplicar un filtro de p-valor ajustado < 0.005, el número de términos se redujo a aproximadamente 25 procesos significativamente enriquecidos. Estos términos incluyen procesos relacionados con el metabolismo del β-amiloide, la inflamación microglial, el procesamiento del receptor Notch y diversas rutas metabólicas y de transporte neuronal.

Para generar las representaciones gráficas, específicamente el diagrama de Venn (Figura 11), fue necesario realizar un análisis funcional adicional empleando el algoritmo Leiden y maximizando el puntaje de enriquecimiento funcional, lo que ha permitido realizar una comparación detallada entre este análisis funcional y el realizado anteriormente, ambos basados en el algoritmo Leiden, pero con diferentes

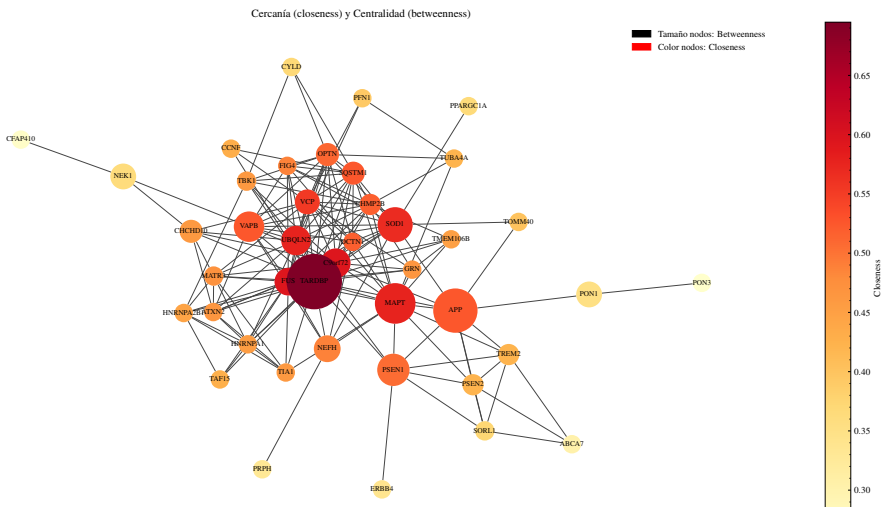

Figure 5: Representación visual de la relación entre closeness (cercanía) y betweenness (centralidad de intermediación) para los nodos de la red. El tamaño de los nodos ha sido establecido en base al valor de centralidad obtenido y el valor de cercanía se establece con el color del nodo.

enfoques el primero para optimizar la modularidad máxima y el segundo para optimizar el puntuaje de enrequecimiento máximo.

A continuación, se muestran representaciones gráficas del análisis funcional realizado tras aplicar el filtro de p-valor ajustado. Estas representaciones incluyen un diagrama de Venn (Figura 11), que permite visualizar las intersecciones entre los conjuntos de genes asociados a diferentes procesos; un gráfico de barras (Figura 12), que muestra los términos significativamente enriquecidos y su nivel de significancia; un gráfico de puntos (Figura 13), que relaciona los términos con su proporción y puntuación de enriquecimiento; y un gráfico de concentración y dispersión (Figura 14), que destaca la distribución de genes clave en los procesos identificados. Estas gráficas ilustran los hallazgos, proporcionando una visión integral tanto de la relevancia estadística de los términos identificados como de las conexiones funcionales entre los genes y los procesos biológicos correspondientes.

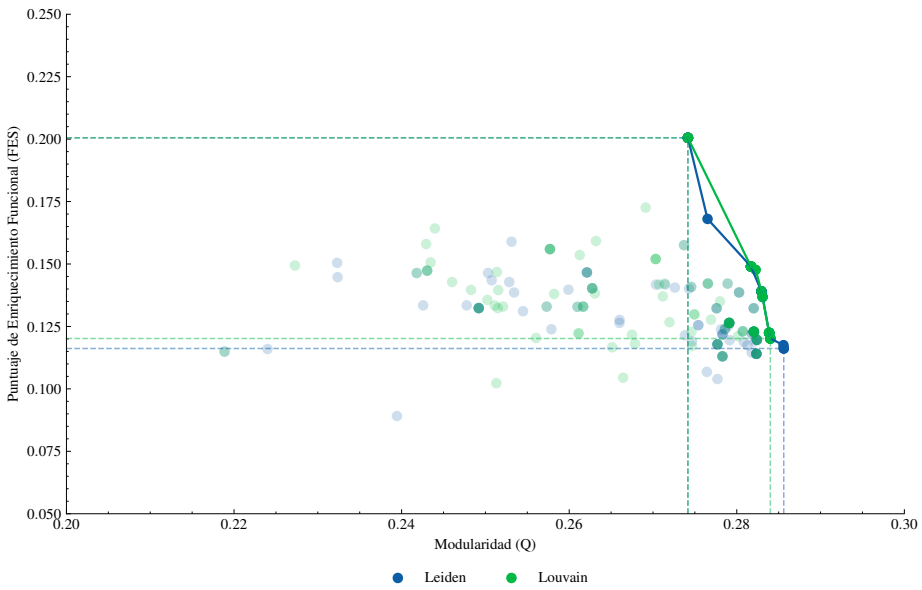

Figure 6: Frente de Pareto obtenido tras maximizar el puntuaje de FES y el coeficiente Q para Leiden y Louvain. Se puede observar que, para las soluciones Pareto-eficientes que conforman el extremo de la frontera, Leiden obtiene un puntuaje superior en cuanto a la modularidad, mientras que ambos algoritmos están igualados en cuanto al puntuaje de enriquecimiento funcional, diferenciándose por pocos decimales.

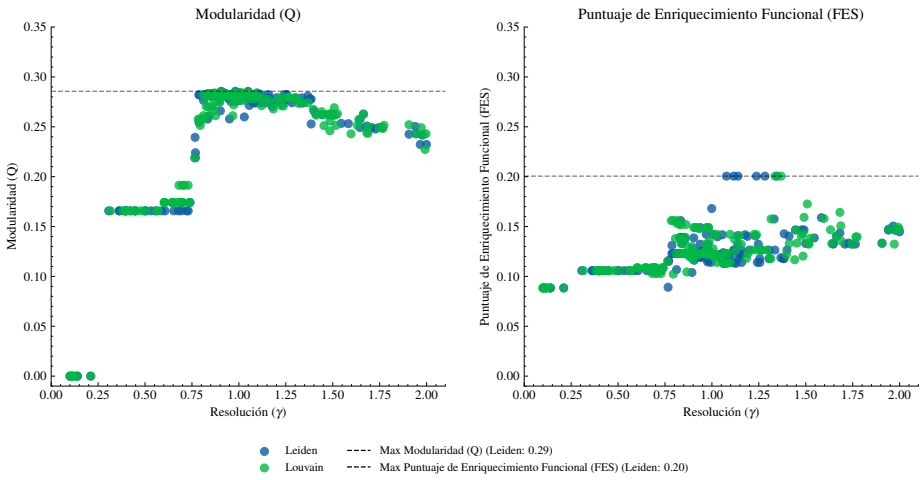

Figure 7: Valores de resolución (γ) explorados por algoritmo, frente al rendimiento evaluado sobre el clustering obtenido. Leiden obtiene por un margen considerable la mejor modularidad, mientras que Louvain obtiene el mejor valor del FES por una diferencia despreciable.

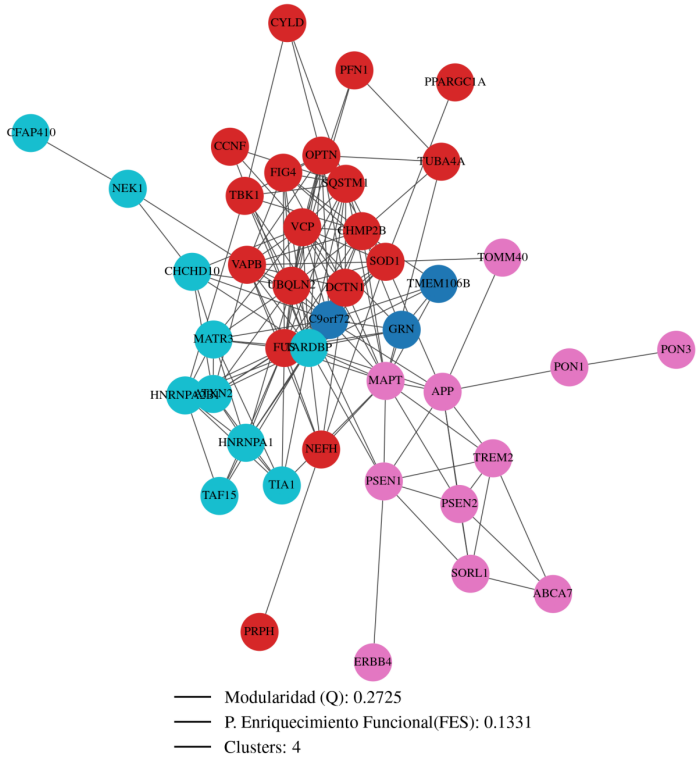

Figure 8: Resultados de clustering para el algoritmo Fast-Greedy. Este clustering será considerado como nuestra base. No se ha ajustado ningún hiperparámetro para este método.

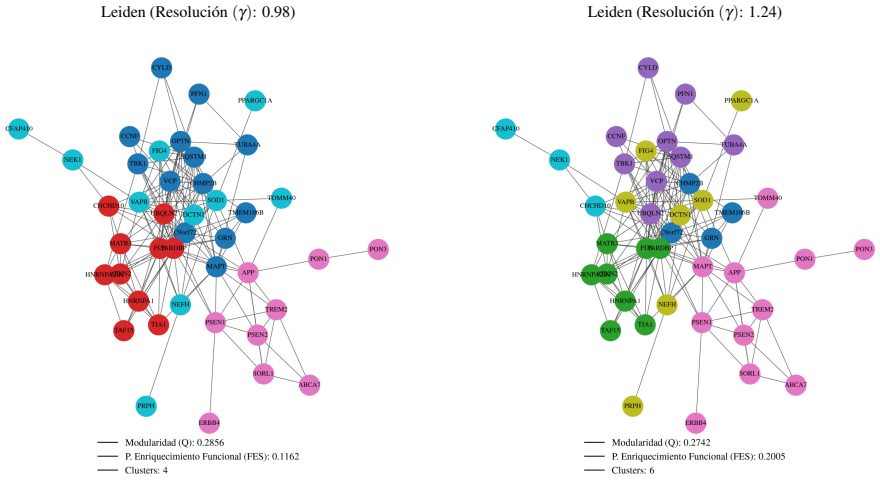

Figure 9: Resultados de clustering para el algoritmo Leiden. Se presentan dos resultados, que corresponden a las soluciones Pareto-óptimas extremas ($max(Q)$ ó $max(FES)$).

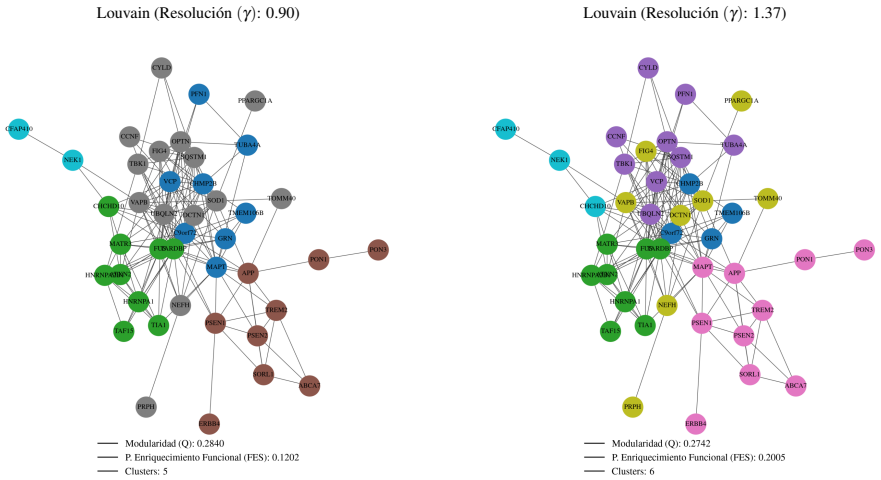

Figure 10: Resultados de clustering para el algoritmo Louvain. Se presentan dos resultados, que corresponden a las soluciones Pareto-óptimas extremas ($max(Q)$ ó $max(FES)$).

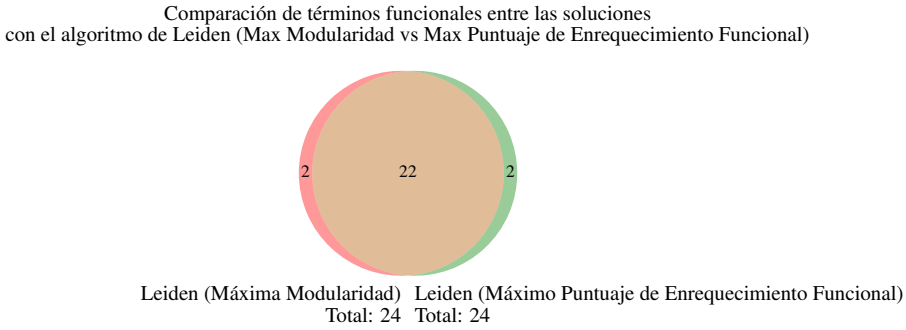

Figure 11: Este diagrama de Venn compara los términos funcionales identificados al aplicar el algoritmo de Leiden utilizando dos criterios: máxima modularidad y máximo puntaje de enriquecimiento funcional. La superposición indica los términos compartidos entre ambas soluciones, lo que proporciona una visión de la robustez y consistencia de los procesos identificados. Esta comparación ayuda a validar los resultados del análisis de clustering y enriquecimiento funcional

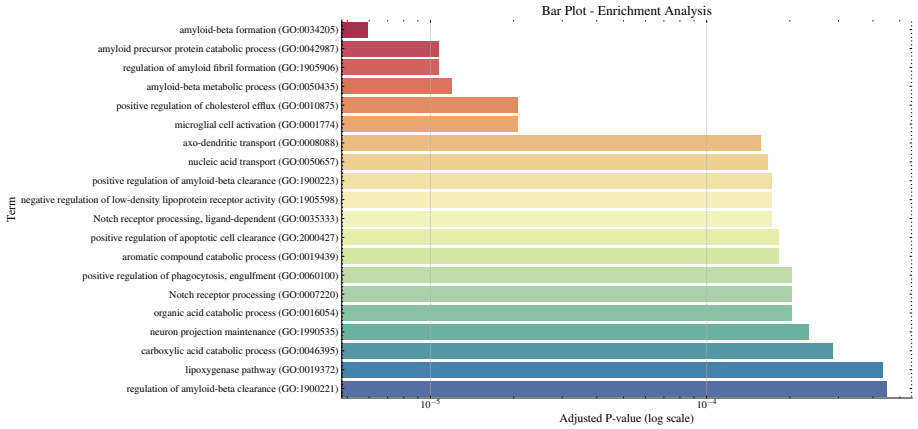

Figure 12: Este gráfico de barras muestra aquellos procesos biológicos enriquecidos del fenotipo de demencia frontotemporal (DFT). Los términos GO (Gene Ontology) más relevantes están ordenados por el p-valor ajustado en una escala logarítmica. La longitud de las barras indica la magnitud del enriquecimiento, mientras que el p-valor ajustado refleja la significancia estadística de cada proceso identificado.

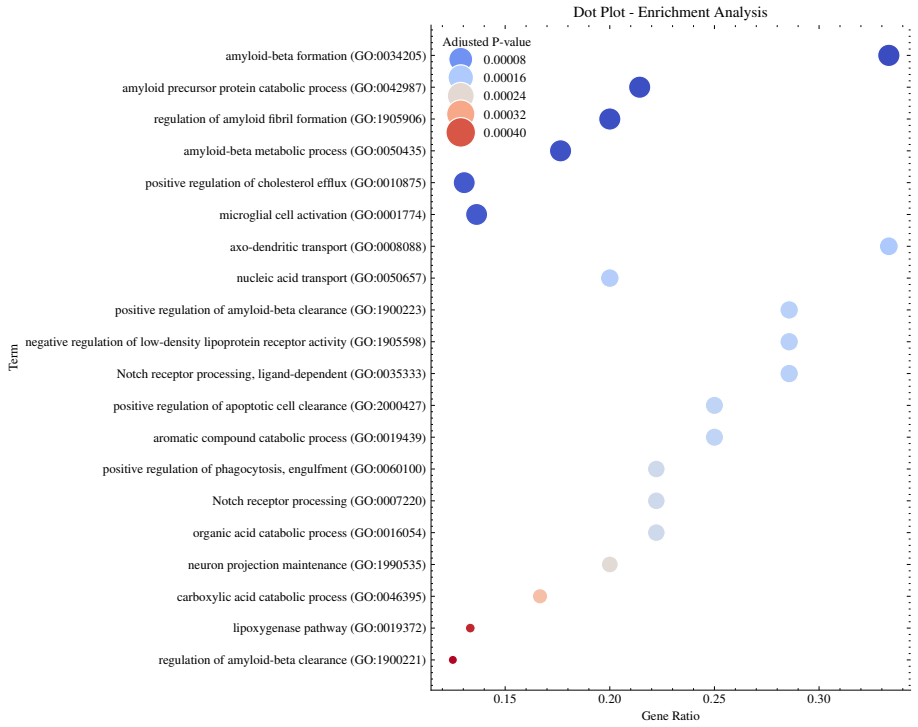

Figure 13: Este gráfico de puntos visualiza los términos GO enriquecidos junto con el Gene Ratio y el p-valor ajustado. El Gene Ratio indica la proporción de genes asociados a cada proceso respecto al total de genes en la lista de entrada. El tamaño del punto representa el número de genes implicados en cada término, mientras que el color refleja el p-valor ajustado.

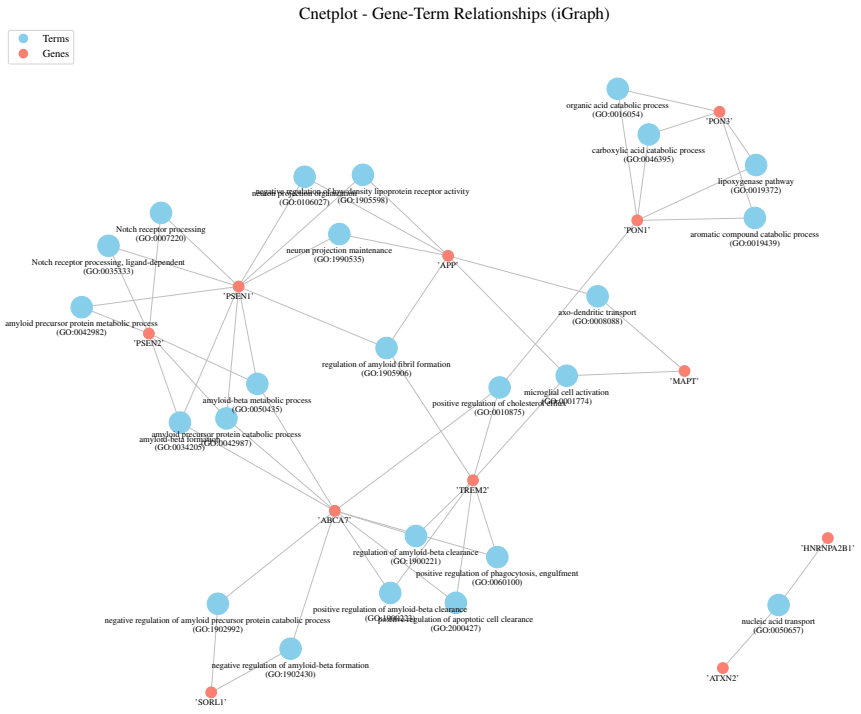

Figure 14: Este gráfico muestra la relación entre genes específicos y los procesos biológicos enriquecidos en la DFT. Cada término GO está conectado con los genes que participan en dicho proceso. Esta visualización permite identificar genes que actúan como puntos de convergencia funcional.

5 Discusión

En este estudio se ha llevado a cabo un análisis de clustering con dos configuraciones del algoritmo de Leiden: una optimizada para máxima modularidad y otra para máximo puntaje de enriquecimiento funcional. Para demostrar que ambas configuraciones producen resultados muy similares, se han comparado los resultados utilizando un diagrama de Venn (Figura 11). Este diagrama destaca los procesos biológicos comunes entre ambas configuraciones, así como los exclusivos de cada enfoque.

Se observa la existencia de 22 términos compartidos entre los 24 identificados para cada configuración del algoritmo de Leiden, lo que refuerza los hallazgos del análisis funcional. Los procesos comunes representan rutas biológicas esenciales que son robustas frente a las variaciones metodológicas, sugiriendo que reflejan aspectos centrales del fenotipo estudiado. Esto no solo valida la calidad de los resultados obtenidos, sino que también subraya la importancia de integrar diferentes enfoques para obtener una visión más completa y confiable del sistema biológico en análisis.

En esta sección, discutiremos los resultados obtenidos del análisis funcional que se realizó únicamente con la configuración de máximo puntaje de enriquecimiento funcional. Dado que, como se ha señalado previamente, el diagrama de Venn (Figura 11) refuerza la confianza de los siguientes hallazgos, ya que muestra una gran similitud con los resultados obtenidos bajo la configuración de máxima modularidad. Por esta razón, la elección de una única configuración asegura una interpretación robusta y confiable.

A partir de los resultados obtenidos y representados mediante gráficos de enriquecimiento (Figura 12, Figura 13 y Figura 14), es posible identificar procesos clave relacionados con la neurodegeneración, la inflamación y el metabolismo neuronal. A continuación, discutiremos los hallazgos más relevantes y su relación con estudios previos, evaluando sus implicaciones para la comprensión de la patogénesis de la DFT.

5.1 Procesos Relacionados con el β-Amiloide

Uno de los hallazgos más destacados en nuestro análisis es la implicación de procesos relacionados con el β-amiloide, como la formación (GO:0034205) y el metabolismo del β-amiloide (GO:0050435). Estos procesos, evidentes en las Figuras 12 y 13, son relevantes por su alta significancia estadística y el número de genes implicados.

Aunque la acumulación de β-amiloide es característica de la enfermedad de Alzheimer, estudios recientes han identificado su presencia en subtipos específicos de DFT y en casos mixtos [**?**, **?**]. La disfunción en el metabolismo del β-amiloide podría contribuir al daño neuronal y a la disfunción sináptica, agravando los síntomas de la DFT [**?**].

Además, procesos de aclaramiento del β-amiloide (GO:1900221) observados en la Figura 14 sugieren fallos en la eliminación de estos péptidos tóxicos [**?**]. La alteración en el aclaramiento puede estar asociada a una activación microglial crónica y a una respuesta inflamatoria exacerbada, que potencian la neurodegeneración [**?**].

Estos resultados resaltan la posible convergencia patogénica entre la EA y la DFT, sugiriendo que modular el metabolismo y aclaramiento del β-amiloide podría ser una estrategia terapéutica a considerar para ciertos subtipos de DFT.

5.2 Inflamación y Activación Microglial

El análisis revela una implicación significativa de procesos relacionados con la activación de células microgliales (GO:0001774) y la regulación positiva de la fagocitosis (GO:0060100), como se muestra en las Figuras 12 y 13. La activación microglial crónica es una característica común en la DFT y otras taupatías, contribuyendo a la progresión de la neurodegeneración [**?**, **?**].

En nuestro estudio, genes como TREM2 destacan en la Figura 14 por su papel central en la activación microglial y la respuesta inmunitaria innata. Mutaciones en TREM2 están asociadas con un mayor riesgo de DFT y otras enfermedades neurodegenerativas, facilitando una respuesta inflamatoria desregulada [**?**, **?**].

La activación prolongada de la microglía puede conducir a la liberación de citocinas proinflamatorias como IL-1β, IL-6 y TNF-*alpha*, lo que exacerba el daño neuronal y sináptico [**?**]. Además, la fagocitosis microglial desregulada puede interferir con el aclaramiento de proteínas tóxicas, contribuyendo a la acumulación de agregados proteicos y promoviendo la neuroinflamación crónica [**?**].

Estos hallazgos sugieren que modular la actividad microglial o reducir la inflamación crónica podría representar una estrategia terapéutica prometedora para mitigar el daño neuronal en la DFT.

5.3 Procesamiento del Receptor Notch

Nuestro análisis identifica procesos relacionados con el procesamiento del receptor Notch (GO:0035333), como se observa en las Figuras 12 y 13. La vía de señalización Notch desempeña un papel fundamental en la diferenciación neuronal, la proliferación celular y el mantenimiento de la homeostasis cerebral [**?**]. La disfunción en esta vía puede comprometer la estabilidad estructural y funcional de las neuronas, contribuyendo a la neurodegeneración observada en la DFT [**?**].

En la Figura 14, genes como PSEN1 y PSEN2 están claramente implicados en estos procesos. Estas proteínas forman parte del complejo de la γ-secretasa, responsable del corte proteolítico del receptor Notch [**?**]. Mutaciones en PSEN1 y PSEN2 no solo afectan el metabolismo del β-amiloide, sino que también interfieren en el procesamiento del receptor Notch, lo que puede llevar a una alteración en la señalización celular y a una disfunción neuronal progresiva [**?**].

Además, una señalización Notch alterada puede impactar negativamente en la neurogénesis y en la capacidad de reparación neuronal, procesos críticos para contrarrestar la neurodegeneración [**?**]. La convergencia de estos mecanismos patológicos sugiere que la disfunción del procesamiento Notch es una característica importante en la patogénesis de la DFT y podría ser un objetivo terapéutico potencial.

5.4 Alteraciones Metabólicas y de Transporte Neuronal

Nuestro análisis identifica procesos relacionados con el transporte axo-dendrítico (GO:0008088) y el transporte de ácidos nucleicos (GO:0050657), representados en las Figuras 13 y 14. Estos procesos son esenciales para el correcto funcionamiento de las neuronas, facilitando la distribución de proteínas, ARNm y organelos a lo largo de los axones y dendritas [**?**].

Las alteraciones en estos mecanismos pueden provocar una acumulación de proteínas mal plegadas y una disfunción sináptica, contribuyendo a la neurodegeneración observada en la DFT [**?**]. En particular, genes asociados con el citosqueleto y motores moleculares (como KIF5A y DYNC1H1) están implicados en estas vías y se destacan en la Figura 14 [**?**].

Además, se observan alteraciones en procesos metabólicos como el catabolismo de compuestos aromáticos (GO:0019439) y el catabolismo de ácidos orgánicos (GO:0016054) en la Figura 12. Estos procesos son críticos para el mantenimiento del equilibrio energético y la eliminación de metabolitos tóxicos [**?**]. La disfunción en el metabolismo neuronal puede generar estrés oxidativo y acumular productos tóxicos, exacerbando la degeneración neuronal en la DFT [**?**].

Estos hallazgos sugieren que las estrategias terapéuticas dirigidas a mejorar el transporte neuronal y modular el metabolismo celular podrían ser beneficiosas para frenar la progresión de la DFT.

5.5 Rol de la Paraoxonasa 1 (PON1)

El análisis identifica a PON1 (Paraoxonasa 1) como un gen clave en los procesos metabólicos y de transporte neuronal, específicamente en el catabolismo de compuestos aromáticos (GO:0019439) y el catabolismo de ácidos orgánicos (GO:0016054), como se observa en la Figura 14. La proteína PON1 participa en la detoxificación de compuestos oxidativos y en el metabolismo de lípidos, contribuyendo a proteger las neuronas del estrés oxidativo [**?**].

La disfunción de PON1 puede conducir a una acumulación de productos tóxicos y a un aumento del daño oxidativo, procesos implicados en la neurodegeneración característica de la DFT [**?**]. Además, estudios sugieren que niveles bajos de actividad de PON1 están asociados con un mayor riesgo de enfermedades neurodegenerativas, debido a su incapacidad para neutralizar los radicales libres [**?**].

La red de interacción proteína-proteína (PPI) muestra que PON1 es un nodo central, lo que sugiere su participación en múltiples rutas metabólicas. Este papel destacado implica que modular la actividad de PON1 podría ser una estrategia terapéutica para reducir el daño oxidativo y mejorar la función metabólica en pacientes con DFT [**?**].

6 Conclusiones

7 Anexo A: KDE mediante ventana deslizante de Parzen-Rosenblatt en el TPE

La Estimación de Densidad por Kernel (KDE) mediante el método de ventana deslizante de Parzen-Rosenblatt es una técnica no paramétrica para estimar la función de densidad de probabilidad (PDF) de una variable aleatoria a partir de una muestra de datos. La estimación en un punto x se realiza sumando las contribuciones de cada dato x_i mediante una función kernel K centrada en x_i:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{6}$$

Donde n es el número de muestras, h es el ancho de banda (parámetro de suavizado) y $K(u)$ es la función kernel, comúnmente el kernel gaussiano:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \tag{7}$$

En el contexto del TPE, este método se utiliza para estimar las distribuciones de probabilidad de los hiperparámetros en los conjuntos de buen rendimiento $\mathcal{C}_1$ y peor rendimiento $\mathcal{C}_2$. Los datos de entrada son los hiperparámetros observados θ_i en cada conjunto, y la salida es una estimación suave de las densidades $l(\theta)$ y $g(\theta)$:

$$\begin{aligned} l(\theta) &= \frac{1}{|\mathcal{C}_1|h} \sum_{\theta_i \in \mathcal{C}_1} K\left(\frac{\theta - \theta_i}{h}\right) \\ g(\theta) &= \frac{1}{|\mathcal{C}_2|h} \sum_{\theta_i \in \mathcal{C}_2} K\left(\frac{\theta - \theta_i}{h}\right) \end{aligned} \tag{8}$$

Al utilizar el método de ventana deslizante de Parzen-Rosenblatt con kernel gaussiano, el TPE obtiene una estimación flexible y no paramétrica de las distribuciones de los hiperparámetros. Esto facilita la exploración eficiente del espacio de búsqueda y la identificación de regiones donde es más probable encontrar hiperparámetros que mejoren el rendimiento del modelo.

Abreviaciones
AD → Alzheimer's Disease (Enfermedad de Alzheimer)
FTD → Frontotemporal Dementia (Demencia Frontotemporal)
bvFTD → Behavioral variant Frontotemporal Dementia (Variante del Comportamiento de la Demencia Frontotemporal)
PPA → Primary Progressive Aphasia (Afasia Progresiva Primaria)
nfvPPA → Nonfluent/Agrammatic variant of PPA (Variante no fluente/agramática de la PPA)
svPPA → Semantic variant of PPA (Variante semántica de la PPA)
lvPPA → Logopenic variant of PPA (Variante logopénica de la PPA)
MND → Motor Neuron Disease (Enfermedad de la Motoneurona)
ELA → Amyotrophic Lateral Sclerosis (Esclerosis Lateral Amiotrófica)
FTLD → Frontotemporal Lobar Degeneration (Degeneración Lobar Frontotemporal)
MRI → Magnetic Resonance Imaging (Imagen por Resonancia Magnética)
PET → Positron Emission Tomography (Tomografía por Emisión de Positrones)
SPECT → Single Photon Emission Computed Tomography (Tomografía por Emisión de Fotón Único)
FDA → Food and Drug Administration (Administración de Alimentos y Medicamentos)
GRN → Granulin precursor gene (Gen precursor de granulina)
GO → Gene Ontology (Gene Ontology)
RQ → Research Question (Preguntas de investigación)
SRQ → Secondary Research Question (Preguntas de investigación secundaria)
FDR → Funtional False Discovery (tasa de descubrimiento falso)
BHO → Bayesian Hyperparameter Optimization (ajuste bayesiano de hiperparámetros)
PDF → Probability Density Function (Función de Densidad de Probabilidad)
KDE → Kernel Density Estimation (Estimación de la PDF mediante Kernel)

Disponibilidad de datos y materiales
https://github.com/MarioPasc/project_template.git

Contribución de los autores
G.M.A : (Redacción) Análisis Genético del Fenotipo, Introducción; Medidas de Rendimiento, Métodos; Algoritmos, Métodos; Flujo de Trabajo, Diagrama. (Código) Metricas de Rendimiento, Modulo Clustering; Algoritmos de Clústering, Modulo Clustering; Visualización del Frente de Pareto y Visualización Ajuste de Hiperparámetros, Modulo Clustering.
C.R.G : (Redacción) Relación del Fenotipo con Enfermedades, Introducción; Análisis de enriquecimiento de vías biológicas, Métodos; Red PPI y sus propiedades, Resultados; (Código) Módulo Network Completo; Obtención Proteínas en utils; Cambio de red .tsv a formato iGraph, utils; Creación del fichero original setup.sh y launch.sh;

A.N.S.B : (Redacción) Definición de Demencia, subtipos, relación con FTD, Introducción; HPO, StringDB, Datos de Entrada, Métodos; Análisis Funcional, Resultados, Discusión; (Código) Módulo Análisis funcional.
M.P.G : (Redacción) Variantes clínicas de la FTD, Introducción; Objetivos, Objetivos; Optimización, Métodos; (Código) Optimización de los algoritmos, módulo clustering; Visualización de los resultados de clústering, módulo clústering; Revisión continua del código; Modificación continua de launch.sh; Despliegue de la imagen Docker.