

RESEARCH

Mitocondrias anormales en el tejido muscular

Julia Del Río Toledo^{*}
, Javier Méndez Parrilla
, Juan Jesús Rus Muñoz
and Benjamín Cortés Fernández

^{*}Correspondence:
juliadelrio2003@uma.es
ETSI Informática, Universidad de
Málaga, Málaga, España
Full list of author information is
available at the end of the article

Abstract

Keywords: sample; article; author

1 Introducción

Las mitocondrias, además de ser la principal fuente de ATP, desempeñan funciones esenciales en la regulación del metabolismo celular, incluyendo la catabolización de nutrientes y la gestión de la homeostasis redox [1]. A lo largo de la evolución eucariota, su forma y comportamiento se adaptaron para garantizar la transmisión precisa de su genoma y responder a las demandas celulares [2]. En tejidos de alta demanda energética como el músculo esquelético, las mitocondrias juegan un papel crucial en la flexibilidad metabólica y la adaptación a estímulos como el ejercicio [3, 4]. La disfunción mitocondrial contribuye a diversas patologías [5], neurodegenerativas, cancerígenas [6] o musculoesqueléticas [7].

La relación entre las mitocondrias y el músculo esquelético es fundamental para comprender el metabolismo energético y la adaptación muscular. La biogénesis mitocondrial y la captación de glucosa estimulada por contracción son procesos clave que se ven influenciados por factores como el óxido nítrico [8] o el OPA1 (Optic Atrophy 1) [9]. Además, el ejercicio regular desencadena adaptaciones moleculares que optimizan el rendimiento muscular y la salud mitocondrial [10, 11]. Sin embargo, con el envejecimiento, la dinámica mitocondrial y la capacidad de mitofagia pueden verse comprometidas, afectando así la funcionalidad muscular [12, 13].

La mitofagia es el proceso mediante el cual las mitocondrias dañadas se eliminan selectivamente. Los defectos en dicho proceso pueden llevar a la acumulación de mitocondrias disfuncionales, afectando aún más la salud muscular. [13] La disfunción mitocondrial [14] en estos tejidos se ha asociado con varias patologías relacionadas con la edad, incluidas la insuficiencia cardíaca y la sarcopenia [15]. La reducción de la función mitocondrial en el músculo esquelético contribuye a la pérdida de masa muscular y fuerza, lo que afecta significativamente la movilidad y la calidad de vida. Otro factor determinante que provocan las mitocondrias disfuncionales es el aumento de ROS (especies reactivas de oxígeno), provocando daño oxidativo en músculos y otros tejidos. [16]

Además, la disrupción en la dinámica mitocondrial (fusión y fisión) contribuye a la atrofia muscular, con un aumento de mitocondrias hinchadas y fragmentadas. Estas alteraciones son comunes en enfermedades como la distrofia muscular, donde

la pérdida de proteínas estructurales agrava los efectos negativos de la disfunción mitocondrial en el tejido muscular.

En concreto, este estudio, examinará los genes vinculados a la patología HP:0008316 (mitocondrias anormales en el tejido muscular) mediante técnicas de biología de sistemas. Nuestro objetivo es investigar, construir y analizar redes biológicas para identificar las vías y procesos relacionados con las enfermedades asociadas a esta condición.

2 Materiales y métodos

2.1 Materiales

2.1.1 *Programas*

R es un lenguaje de programación y un entorno estadístico ampliamente utilizado en bioinformática, análisis de datos y ciencia de datos. Sirve para realizar análisis estadísticos complejos, visualización de datos y manipulación de datos biológicos, como el análisis de expresión genética, que resulta relevante en este estudio. Fue desarrollado por Robert Gentleman y Ross Ihaka en la Universidad de Auckland, Nueva Zelanda, y es un proyecto de código abierto mantenido por la comunidad R [17].

En este trabajo, se utilizó la versión 4.4.2 de R, que incluye mejoras recientes en rendimiento y funcionalidad. R ha logrado un lugar destacado en bioinformática y ciencia de datos gracias a su flexibilidad y extensibilidad, permitiendo abordar desde análisis de datos genómicos hasta visualización de redes biológicas complejas. Esto se ha reflejado en su amplia adopción y en la disponibilidad de paquetes especializados que simplifican tareas como el análisis de secuencias, la estadística avanzada y la integración de múltiples tipos de datos biológicos [18].

En este proyecto, R se empleó como una herramienta integral para el análisis de redes y la representación funcional de datos genómicos, haciendo uso de técnicas computacionales modernas y métodos estadísticos reproducibles. Su capacidad para integrar datos biológicos complejos lo convierte en un recurso invaluable en investigaciones bioinformáticas actuales [19].

Python es un lenguaje de programación ampliamente utilizado en bioinformática y análisis de datos biológicos debido a su simplicidad y versatilidad. Es especialmente útil para gestionar y analizar grandes volúmenes de datos genómicos, así como para automatizar flujos de trabajo. En este estudio, se empleó Python para interactuar con bases de datos biológicas y obtener información sobre genes y sus interacciones proteicas, utilizando su capacidad para integrar diferentes tecnologías y bibliotecas.

La versión 3.12.0 de Python se utilizó en este proyecto, aprovechando las mejoras en rendimiento y manejo de memoria, lo que resulta esencial cuando se trabajan con grandes cantidades de datos. Este lenguaje ha demostrado ser clave en estudios similares, donde se utiliza para realizar análisis de redes proteicas y facilitar la reproducibilidad de los resultados científicos [20].

2.1.2 *Bases de Datos*

HPO (Human Phenotype Ontology) es una base de datos que proporciona una clasificación estándar de los términos fenotípicos humanos, cada uno con un identificador único. En el contexto de la investigación genética, HPO permite relacionar

los fenotipos observables ("mitocondrias anormales en el tejido muscular", identificado por HP:0008316) con genes y enfermedades específicas. Esto facilita el análisis y busca de datos genéticos en función de características fenotípicas. Es mantenido por la comunidad científica, específicamente por el equipo de Monarch Initiative, para mejorar la precisión en los estudios de fenotipos y genética humana.[21]

StringDB es una base de datos que contiene información sobre interacciones proteína-proteína (PPI), recopilada de diversas fuentes como experimentos, bases de datos públicas y predicciones computacionales. Esta base de datos permite investigar redes de proteínas interconectadas, facilitando el análisis de cómo las proteínas interactúan en diferentes contextos biológicos, algo esencial en estudios genéticos como los relacionados con HPO. Desarrollada por un consorcio de investigadores y está disponible como un recurso en línea gratuito para la comunidad científica.[22]

El **NCBI** (National Center for Biotechnology Information) es un centro de investigación en biotecnología de Estados Unidos que proporciona acceso a una vasta colección de bases de datos biológicas y genómicas. Es utilizado para acceder a datos sobre genes, proteínas y secuencias genómicas, y proporciona herramientas de análisis. Nosotros lo utilizamos junto a HPO, para facilitar la correlación entre genes específicos y fenotipos. Fue fundado por los Institutos Nacionales de Salud (NIH) de Estados Unidos y es un recurso fundamental para la investigación genética y bioinformática.[23]

2.1.3 Bibliotecas

Bibliotecas en R

El análisis y la visualización de datos en este estudio se ha realizado utilizando varias bibliotecas de R, que proporcionan herramientas especializadas para el manejo de redes, análisis de genes y visualización de resultados. A continuación, se describen las principales bibliotecas utilizadas.

Igraph (versión 2.1.2) es una biblioteca para el análisis y la visualización de redes complejas, desarrollada en múltiples lenguajes, incluyendo Python y R. En bioinformática, se utiliza para construir y analizar redes de interacciones biológicas, como redes de proteínas o genes, ayudando a visualizar cómo las proteínas relacionadas con un fenotipo específico pueden interactuar entre sí. Esta biblioteca es especialmente útil para el análisis de grafos de interacciones, permitiendo obtener métricas como la centralidad, coeficiente de agrupamiento y otros atributos topológicos de las redes. Fue desarrollada por una comunidad internacional de programadores y científicos[24] [25] [26].

ClusterProfiler (versión 4.14.4) es una biblioteca utilizada para realizar análisis de enriquecimiento funcional de genes. Esta herramienta permite identificar las rutas biológicas, funciones moleculares y procesos celulares que están significativamente representados en los conjuntos de genes estudiados. En este trabajo, se empleó para realizar el enriquecimiento de los genes asociados a interacciones de proteínas y a fenotipos específicos. [27] [28] [29] [30].

Org.Hs.eg.db (versión 3.20.0) proporciona una base de datos extensa con información sobre genes humanos, incluyendo símbolos de genes, identificadores de

acceso, y asociaciones con términos GO, entre otras. Esta base de datos fue crucial para mapear los genes a los términos correspondientes en los análisis de enriquecimiento. [31]

Enrichplot (versión 1.26.3) se utilizó para la representación gráfica de los resultados de los análisis de enriquecimiento. Esta biblioteca facilita la creación de visualizaciones, como gráficos de barras, gráficos de puntos y diagramas de enriquecimiento, para facilitar la interpretación de los datos. [32]

Optparse (versión 1.7.5) es una biblioteca que se utilizó para implementar interfaces de línea de comandos en los scripts de Python, permitiendo a los usuarios especificar parámetros de entrada y salida de manera flexible y eficiente. [33]

GGplot2 (versión 3.5.1) es una biblioteca popular para la creación de gráficos en R. En este estudio, se utilizó para generar visualizaciones adicionales de los resultados de los análisis y facilitar la interpretación visual de las interacciones de proteínas y los enriquecimientos funcionales. [34]

Bibliotecas en Python

Pandas es una biblioteca en Python que facilita la manipulación y el análisis de datos mediante estructuras de datos como DataFrames. En estudios bioinformáticos, pandas se utiliza para manejar y procesar grandes volúmenes de datos, como los obtenidos de bases de datos genéticas. Es útil para organizar los datos de HPO y facilitar su análisis. Fue creada por Wes McKinney y es ahora una biblioteca ampliamente adoptada en el ámbito de la ciencia de datos. [35]

Requests es una biblioteca de Python que permite realizar solicitudes HTTP de forma sencilla. Es utilizada para acceder a APIs y extraer datos de bases de datos en línea, como HPO o StringDB, permitiendo automatizar la obtención de datos relevantes para el estudio de fenotipos y genes. Fue desarrollada por Kenneth Reitz y es una de las bibliotecas más populares para trabajar con APIs en Python. [36]

Networkx es una biblioteca en Python para la creación, manipulación y análisis de redes complejas. Se utiliza para modelar y analizar redes de interacciones genéticas o de proteínas. Por ejemplo, en el análisis de datos de StringDB, networkx permite estudiar las relaciones entre proteínas de forma visual y cuantitativa. Es una biblioteca de código abierto mantenida por la comunidad, utilizada ampliamente en estudios de redes biológicas y sociales. [37]

Matplotlib.pyplot es una biblioteca de Python que permite crear gráficos y visualizaciones de datos de forma detallada y personalizada. Lo utilizamos para representar datos biológicos y resultados de análisis de manera visual, lo cual facilita la interpretación de datos complejos, como los patrones de interacción entre proteínas o genes relacionados con un fenotipo específico. Fue desarrollada por John D. Hunter y es mantenida por la comunidad científica y de datos. [38]

2.2 Métodos

2.2.1 Obtención de datos y construcción de la red de interacciones

El análisis comenzó con la identificación de genes asociados al fenotipo de interés, definido por el código HP:0008316, que corresponde a "mitocondrias anormales en tejido muscular" en la Ontología del Fenotipo Humano (HPO, por sus siglas en

inglés). Este paso fue crucial para establecer una base de datos inicial que sustentara el estudio.

Para llevar a cabo este análisis de manera sistemática, desarrollamos un script en **Python** llamado *hpo_genes_fetcher.py* que interactuaba con la API de **HPO**. La elección de Python como lenguaje principal se basó en su flexibilidad y en su amplio ecosistema de bibliotecas, como *requests* y *pandas*, que son ideales para manejar datos y realizar solicitudes con las APIs. El script se diseñó para automatizar la extracción de genes relacionados con el fenotipo, lo que evitaba errores manuales y garantizaba la reproducibilidad del proceso.

Se utilizó la biblioteca *requests* para realizar solicitudes HTTP de tipo GET a los endpoints específicos de la API de HPO. El identificador del fenotipo, HP:0008316, se incluyó como parámetro en la URL de la solicitud, asegurando que los datos recuperados estuvieran específicamente relacionados con este término fenotípico. La respuesta de la API fue procesada y transformada. Esta contenía una lista dentro del campo "genes", que incluía los nombres de los genes asociados al fenotipo.

Finalmente, los genes se almacenaron en un archivo de texto estructurado llamado *genes_list.txt*, dentro de un directorio especificado, para facilitar su acceso y uso en las etapas siguientes del estudio.

El archivo generado fue utilizado como entrada en un segundo script de Python, *fetch_interactions.py*, que se encargó de analizar las interacciones proteicas entre las proteínas codificadas por los genes recopilados. Este análisis es crucial para comprender cómo interactúan entre sí las proteínas codificadas por los genes, lo que puede revelar información sobre sus funciones en el fenotipo estudiado.

Utilizando la API de **STRINGdb**, el script empleó solicitudes POST para enviar los nombres de las proteínas. Se configuraron parámetros como la especie de interés, en nuestro caso el Homo sapiens, cuyo código en el **NCBI** es 9606; y un umbral de confianza, aunque se encontraba por defecto con un valor de 0.5, se ejecutó con 0.9, significando esto que las interacciones incluidas tienen al menos un 90% de probabilidad de ser funcionalmente relevantes, según los cálculos de STRINGdb. Este valor más restrictivo aseguró que únicamente se consideraran interacciones de alta confiabilidad, aumentando así la precisión de los resultados.

La respuesta de la API, en formato JSON, proporcionó información detallada sobre las interacciones, incluyendo los genes involucrados, la naturaleza de las interacciones y una puntuación de confianza. Estos datos se filtraron y almacenaron en un archivo CSV, *interacciones_proteinas.csv*, lo que permitió un análisis estructurado y reproducible.

Además, el script incluyó una funcionalidad para generar una representación visual de la red de interacciones en formato PNG, utilizando otro endpoint de STRINGdb. Esta imagen, guardada en un directorio específico, facilitó la documentación y presentación gráfica de los resultados, ayudando a comprender la topología de la red y la conectividad de las proteínas involucradas.

En conjunto, estos dos scripts establecieron las bases para el análisis detallado de las interacciones proteicas y su papel potencial en el fenotipo estudiado, integrando datos de diversas fuentes de manera automatizada y reproducible.

2.2.2 Análisis de la red y visualización avanzada

El análisis detallado de la red de interacciones proteicas se realizó en **R** utilizando la biblioteca **iGraph**. Este paso fue fundamental para profundizar en las propiedades estructurales y funcionales de la red, permitiendo identificar patrones clave y relaciones críticas entre los genes asociados a nuestro fenotipo. Los datos generados previamente con STRINGdb, que contenían la lista de nodos y aristas en formato CSV, fueron importados en R mediante la función `read.csv()`, asegurando un flujo de trabajo integrado y reproducible entre las herramientas empleadas.

Los datos se transformaron en un grafo no dirigido utilizando `graph_from_data_frame()`, donde los nodos representaban genes y las aristas sus interacciones proteicas. Este grafo permitió analizar y visualizar las conexiones de la red, asignando propiedades como colores, tamaños y pesos según métricas específicas. Aunque las interacciones se procesaron como no dirigidas, los pesos de las aristas correspondieron a la puntuación de confianza calculada por STRINGdb, lo que proporcionó un marco robusto para interpretar la importancia funcional de cada conexión.

Visualización de la red

Para la visualización, se utilizaron diversas estrategias para representar gráficamente la red, destacando características clave mediante personalizaciones avanzadas en iGraph. La función `plot()` se empleó para crear visualizaciones iniciales, ajustando los colores y tamaños de los nodos en función de su grado de conectividad, lo que permitió identificar fácilmente los genes más conectados o "hubs". Además, se probaron diferentes algoritmos de disposición, como:

- `layout_with_fr` (Fruchterman-Reingold), ideal para visualizar la red como un sistema de fuerzas equilibradas.
- `layout_in_circle`, para destacar patrones circulares de conectividad.

Estas visualizaciones se generaron y guardaron automáticamente en archivos PNG utilizando el script ***plot_network.R***. Este produjo representaciones visuales avanzadas, incluyendo: redes básicas para obtener una visión general de la topología, redes escaladas por grado de conectividad de los nodos y mapas de calor que ilustraron la intensidad de las interacciones entre genes.

Además de las visualizaciones, se calcularon un conjunto de métricas clave para describir la red, incluyendo:

- Grado de conectividad: Para identificar genes con mayor número de interacciones (hubs).
- Centralidad de intermediación (betweenness): Para detectar nodos clave en el flujo de información.
- Coeficiente de agrupamiento: Para evaluar la formación de clústeres densos de genes.
- Densidad de la red y conectividad global: Indicando el nivel general de cohesión de la red.

Estas métricas se resumieron en un archivo de texto (*network_metrics.txt*), que sirvió como referencia para el análisis estructural.

Análisis de comunidades

Por otra parte, la identificación y análisis de comunidades funcionales se llevó a cabo utilizando el script ***cluster_proteinas.R***, que implementaba varios algoritmos de agrupamiento como:

- Edga Betweenness: Basado en la eliminación de aristas críticas.
- Walktrap: Basado en recorridos aleatorios para identificar módulos densos.
- Fast Greedy, Infomap y Label Propagation: Métodos optimizados para diferentes contextos topológicos.

El script seleccionó automáticamente el mejor algoritmo en función de la modularidad, una métrica que mide la calidad de los clústeres formados. Las comunidades detectadas se visualizaron en gráficos coloreados y dendogramas, destacando la jerarquía de los clústeres y los nodos principales en cada uno. Los genes pertenecientes a cada comunidad se guardaron en archivos separados para facilitar su análisis funcional posterior.

La integración de métricas, visualizaciones y análisis de comunidades permitió identificar subgrupos funcionales dentro de la red de interacción proteica. Cada clúster se analizó considerando su relevancia en el contexto del fenotipo, proporcionando información valiosa sobre las relaciones biológicas subyacentes. Este enfoque combinó análisis cuantitativo y cualitativo, ofreciendo una interpretación más profunda y completa de los datos.

2.2.3 Análisis de enriquecimiento funcional

El análisis de enriquecimiento funcional se llevó a cabo utilizando el enfoque **Over-Representation Analysis (ORA)**, que se basa en identificar términos funcionales significativamente sobrerrepresentados en un conjunto de genes en comparación con un fondo genómico de referencia. Este método fue implementado en R mediante el script ***functional_analysis.R***, diseñado para procesar múltiples conjuntos de genes de forma automatizada. Se emplearon herramientas como **clusterProfiler** y **org.Hs.eg.db**, ampliamente reconocidas en bioinformática por su robustez y versatilidad en el análisis funcional.

El script fue configurado para trabajar con directorios de entrada y salida definidos por el usuario a través de argumentos en la línea de comandos. El directorio de entrada contenía archivos de texto con listas de genes que habían sido identificados y agrupados previamente. Cada archivo representaba un conjunto de genes independiente, lo que permitió evaluar términos funcionales específicos para cada comunidad detectada durante el análisis de la red. Los archivos de salida generados consistieron en gráficos en formato PNG que resumían los resultados obtenidos para cada conjunto analizado.

En el primer paso del script, las listas de genes fueron leídas mediante la función *readLines()* y procesadas con *trimws()* para eliminar espacios en blanco, asegurando que los identificadores fueran válidos y consistentes con las bases de datos utilizadas. El formato de los genes como símbolos estándar (*SYMBOL*) garantizó su compatibilidad con las bases de datos GO (Gene Ontology). Estos identificadores fueron utilizados directamente en la función *enrichGO()* de *clusterProfiler*, la cual permitió realizar el análisis de enriquecimiento de procesos biológicos (GO-BP).

El análisis ORA aplicado evaluó si los genes de cada conjunto estaban sobrerrepresentados en términos funcionales específicos de GO-BP. La configuración de *enrichGO()* incluyó parámetros como el ajuste del p-valor mediante el método de Benjamini-Hochberg (BH), que controla la tasa de falsos positivos en análisis con múltiples pruebas. También se estableció un umbral de significancia de 0.05 para incluir únicamente los términos más relevantes. El análisis se basó en la comparación de los genes proporcionados frente a un fondo genómico estándar, permitiendo identificar términos enriquecidos que resaltan procesos biológicos clave, como la regulación del metabolismo energético y la dinámica mitocondrial, ambos directamente relacionados con el fenotipo HP:0008316.

Los resultados del análisis se visualizaron mediante gráficos de puntos (dotplot) generados con las herramientas de visualización de clusterProfiler y **ggplot2**. En estos gráficos, cada punto representaba un término enriquecido de GO-BP, donde su tamaño indicaba el grado de enriquecimiento (proporción de genes asociados) y su posición en el eje reflejaba la significancia estadística. Los gráficos fueron guardados automáticamente en el directorio de salida especificado, con nombres que reflejaban el conjunto de genes analizado. Esto aseguró una organización clara y reproducible de los resultados.

Se eligió ORA como el enfoque principal de enriquecimiento funcional debido a la naturaleza de nuestros datos, que consistían en listas específicas de genes agrupados. Otros enfoques, como GSEA o análisis basados en datos de expresión diferencial, no eran adecuados ya que no disponíamos de datos de expresión o variaciones genéticas asociadas al fenotipo. La decisión de emplear ORA estuvo respaldada por su capacidad para proporcionar información funcional relevante en contextos donde las listas de genes ya están definidas.

En resumen, el análisis ORA implementado mediante este script automatizado permitió identificar procesos biológicos y rutas funcionales clave asociados a los genes agrupados en clústeres, proporcionando una perspectiva funcional complementaria al análisis topológico de la red.

3 Resultados

En el marco de este estudio, se obtuvieron inicialmente 48 genes asociados con el fenotipo mitocondrial desde la base de datos Human Phenotype Ontology (HPO), que se relacionan con funciones celulares esenciales en la mitocondria, como la cadena respiratoria, la biogénesis mitocondrial, y la regulación del metabolismo energético. Estos genes fueron seleccionados con el objetivo de explorar su posible interrelación dentro de una red de interacciones de proteínas, utilizando la base de datos STRING.

Al realizar el análisis de interacciones utilizando los identificadores de estos 48 genes, se observó que solo 38 genes interaccionaron entre sí dentro de la red generada. Este hallazgo sugiere que, aunque todos los genes descargados están asociados con el fenotipo mitocondrial, no todos ellos tienen interacciones directas en el contexto de las proteínas mitocondriales. En particular, algunos genes no mostraron evidencia de interacción directa con otros genes seleccionados, lo que podría indicar que sus roles en la biología mitocondrial pueden no depender de interacciones proteicas directas o que interactúan a través de mecanismos más complejos que no son reflejados en esta red.

Uno de los genes más relevantes en este análisis es MYH7, que está relacionado con la miocardiopatía y es conocido por su papel en la función mitocondrial. Sin embargo, MYH7 no presentó interacciones con otros genes mitocondriales en esta red, lo que sugiere que su influencia en el fenotipo mitocondrial podría estar mediada por mecanismos independientes, tal vez a través de la regulación de procesos mitocondriales o por interacciones con otras proteínas no presentes en la red de interacciones utilizadas en este análisis.

Para profundizar en el análisis de la red obtenida, se han aplicado diferentes algoritmos de clustering con el objetivo de identificar comunidades o grupos de proteínas que interactúen más estrechamente entre sí. Los algoritmos utilizados fueron: Fast Greedy, Edge Betweenness, Walktrap, Infomap y Label Propagation, y se evaluaron mediante la métrica de modularidad, que mide la calidad de las particiones realizadas.

Algoritmo	Modularidad	Número de Clusters
Fast Greedy	0.06917908	3
Walktrap clustering	0.02946764	9
Edge Betweenness clustering	0.01066141	1
Infomap clustering	2.220446×10^{-16}	1
Label Propagation clustering	2.220446×10^{-16}	1

Entre los algoritmos evaluados observamos que Fast Greedy, destacó como el algoritmo más efectivo, lo que indica que logra separar de forma significativa las diferentes comunidades.

El algoritmo identificó tres clúster principales:

- En el primer clúster podemos encontrar nodos como (NDUFS7, NDUFAP1, NDUFB10, NDUFS6, NDUFV2), este clúster reúne muchas proteínas de familia NDUF, fundamentales para la producción de energía en la célula. Entre estos elementos hay una alta conectividad.

- El segundo clúster contiene como nodos más destacados (MT-CO1, MT-ATP6, MT-ND6, MT-ND5), este grupo parece estar más enfocado en la respiración celular y fosforilación oxidativa.

- El tercer clúster, está representado por elementos menos conectados como por ejemplo: NDUFS2, TIMMDC1, NUBPL. Este clúster podría indicar que se están enfocando en procesos relacionados con la cadena de transporte de electrones.

El dendrograma generado mediante clustering jerárquico aportó una perspectiva adicional sobre cómo se relacionaban los genes seleccionados. Esta herramienta organizó los genes en una estructura jerárquica basada en sus similitudes funcionales, permitiendo observar con mayor claridad la proximidad o separación entre ellos dentro de la red de interacciones.

Agrupación de genes

En las ramas inferiores del dendrograma, destacaron genes como **NDUFS7**, **NDUFS3** y **NDUFS6**, los cuales presentaban una alta conectividad en la red de interacciones y se agrupaban estrechamente. Esto reforzó los hallazgos de clustering realizados previamente, donde estos genes se identificaron como parte de

comunidades definidas por algoritmos como Fast Greedy. Su proximidad sugirió roles interrelacionados en la producción de energía celular.

Separación de grupos

A medida que se ascendía en la jerarquía, genes como **NDUFB10** y **MT-CO1** comenzaban a divergir, formando grupos más distantes. Este patrón era consistente con los resultados del algoritmo Walktrap, donde dichos genes pertenecían a clusters separados, lo que indicaba funciones menos directamente conectadas dentro de la red.

Genes periféricos

En las ramas más alejadas del dendrograma se encontraban genes como **NUBPL** y **TIMMDC1**, los cuales presentaban menor conectividad con los clusters principales. Sin embargo, estos genes podrían desempeñar roles complementarios, interactuando de manera más indirecta con genes centrales como **MT-ATP6**.

Relación entre clusters

La estructura jerárquica observada en el dendrograma confirmó la organización de los clusters identificados previamente. Los genes del primer y segundo cluster mostraron una conexión más estrecha, con roles críticos en la cadena respiratoria y la fosforilación oxidativa. Por otro lado, los genes del tercer cluster aparecieron más tempranamente segregados, sugiriendo funciones complementarias o especializadas dentro del contexto mitocondrial.

4 Discusión

5 Conclusiones

Abreviaciones

Indicar lista de abreviaciones mostrando cada acrónimo a que corresponde

Disponibilidad de datos y materiales

[Enlace al proyecto en GitHub](#)

Contribución de los autores

Usando las iniciales que habéis definido al comienzo del documento, debeis indicar la contribución al proyecto en el estilo: J.E : Encargado del análisis de coexpresión con R, escritura de resultados; J.R.S : modelado de red con python y automatizado del código, escritura de métodos; ... OJO: que sea realista con los registros que hay en vuestros repositorios de github.

Author details

ETSI Informática, Universidad de Málaga, Málaga, España.

References

1. Spinelli, J.B., Haigis, M.C.: The multifaceted contributions of mitochondria to cellular metabolism (2018). doi:[10.1038/s41556-018-0124-1](#)
2. Friedman, J.R., Nunnari, J.: Mitochondrial form and function (2014). doi:[10.1038/nature12985](#)
3. Memme, J.M., Erlich, A.T., Phukan, G., Hood, D.A.: Exercise and mitochondrial health. *Journal of Physiology* **599** (2021). doi:[10.1113/JP278853](#)
4. Smith, J.A.B., Murach, K.A., Dyar, K.A., Zierath, J.R.: Exercise metabolism and adaptation in skeletal muscle (2023). doi:[10.1038/s41580-023-00606-x](#)
5. Quintana-Cabrera, R., Scorrano, L.: Determinants and outcomes of mitochondrial dynamics (2023). doi:[10.1016/j.molcel.2023.02.012](#)
6. Chan, D.C.: Mitochondrial dynamics and its involvement in disease. *Annual Review of Pathology: Mechanisms of Disease* **15** (2020). doi:[10.1146/annurev-pathmechdis-012419-032711](#)

7. Liu, S.Z., Marcinek, D.J.: Skeletal muscle bioenergetics in aging and heart failure. *Heart Failure Reviews* **22** (2017). doi:[10.1007/s10741-016-9586-z](https://doi.org/10.1007/s10741-016-9586-z)
8. McConell, G.K., Wadley, G.D.: Potential role of nitric oxide in contraction-stimulated glucose uptake and mitochondrial biogenesis in skeletal muscle. In: *Clinical and Experimental Pharmacology and Physiology*, vol. 35 (2008). doi:[10.1111/j.1440-1681.2008.05038.x](https://doi.org/10.1111/j.1440-1681.2008.05038.x)
9. Noone, J., O’Gorman, D.J., Kenny, H.C.: OPA1 regulation of mitochondrial dynamics in skeletal and cardiac muscle (2022). doi:[10.1016/j.tem.2022.07.003](https://doi.org/10.1016/j.tem.2022.07.003)
10. Hargreaves, M., Spriet, L.L.: Skeletal muscle energy metabolism during exercise (2020). doi:[10.1038/s42255-020-0251-4](https://doi.org/10.1038/s42255-020-0251-4)
11. Egan, B., Zierath, J.R.: Exercise metabolism and the molecular regulation of skeletal muscle adaptation (2013). doi:[10.1016/j.cmet.2012.12.012](https://doi.org/10.1016/j.cmet.2012.12.012)
12. Hood, D.A., Memme, J.M., Oliveira, A.N., Triolo, M.: Maintenance of Skeletal Muscle Mitochondria in Health, Exercise, and Aging (2019). doi:[10.1146/annurev-physiol-020518-114310](https://doi.org/10.1146/annurev-physiol-020518-114310)
13. J, L.-G.: Mitochondrial dynamics and mitophagy in skeletal muscle health and aging. *International Journal of Molecular Sciences* **22** (2021). doi:[10.3390/ijms22158179](https://doi.org/10.3390/ijms22158179)
14. Chen, T.H.: Mitochondrial dysfunction as an underlying cause of skeletal muscle disorders. *International Journal of Molecular Sciences* **23** (2022). doi:[10.3390/ijms232112926](https://doi.org/10.3390/ijms232112926)
15. Boengler, K.: Mitochondria and ageing: role in heart, skeletal muscle and adipose tissue. *Journal of Cachexia, Sarcopenia and Muscle* **8** (2017). doi:[10.1002/jcsm.12178](https://doi.org/10.1002/jcsm.12178)
16. Javadov, S.: Mitochondria-targeted antioxidant preserves contractile properties and mitochondrial function of skeletal muscle in aged rats. *Oncotarget* **6** (2015). doi:[10.18632/oncotarget.5783](https://doi.org/10.18632/oncotarget.5783)
17. Jiménez, J.U.: Introducción a r y rstudio (2019)
18. Giorgi, F.M., Ceraolo, C., Mercatelli, D.: The R Language: An Engine for Bioinformatics and Data Science (2022). doi:[10.3390/life12050648](https://doi.org/10.3390/life12050648)
19. Team, R.C.: R: A language and environment for statistical computing. (2019)
20. Teams, P.: What is Python? Executive Summary (2023)
21. Köhler, S., Doelken, S.C., Mungall, C.J., Bauer, S., Firth, H.V., Bailleul-Forestier, I., Black, G.C., Brown, D.L., Brudno, M., Campbell, J., *et al.*: The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic acids research* **42**(D1), 966–974 (2014)
22. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K.P., *et al.*: String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research* **43**(D1), 447–452 (2015)
23. Jenuth, J.P.: The ncbi: Publicly available tools and resources on the web. *Bioinformatics methods and protocols*, 301–312 (1999)
24. Valdez, B.: Análisis de grafos usando r e igraph. *altamira* **1**(1), 1 (2016)
25. Csardi, G., Nepusz, T.: The igraph software package for complex network research. *InterJournal Complex Systems*, 1695 (2006)
26. Csárdi, G., Nepusz, T., Traag, V., Horvát, S., Zanini, F., Noom, D., Müller, K.: igraph: Network Analysis and Visualization in R. (2024). doi:[10.5281/zenodo.7682609](https://doi.org/10.5281/zenodo.7682609). R package version 2.1.2. <https://CRAN.R-project.org/package=igraph>
27. Yu, G.: Thirteen years of clusterprofiler. *The Innovation* **5**(6), 100722 (2024). doi:[10.1016/j.xinn.2024.100722](https://doi.org/10.1016/j.xinn.2024.100722)
28. Xu, S., Hu, E., Cai, Y., Xie, Z., Luo, X., Zhan, L., Tang, W., Wang, Q., Liu, B., Wang, R., Xie, W., Wu, T., Xie, L., Yu, G.: Using clusterprofiler to characterize multiomics data. *Nature Protocols* **19**(11), 3292–3320 (2024). doi:[10.1038/s41596-024-01020-z](https://doi.org/10.1038/s41596-024-01020-z)
29. Wu, T., Hu, E., Xu, S., Chen, M., Guo, P., Dai, Z., Feng, T., Zhou, L., Tang, W., Zhan, L., xiaochong Fu, Liu, S., Bo, X., Yu, G.: clusterprofiler 4.0: A universal enrichment tool for interpreting omics data. *The Innovation* **2**(3), 100141 (2021). doi:[10.1016/j.xinn.2021.100141](https://doi.org/10.1016/j.xinn.2021.100141)
30. Yu, G., Wang, L.-G., Han, Y., He, Q.-Y.: clusterprofiler: an r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology* **16**(5), 284–287 (2012). doi:[10.1089/omi.2011.0118](https://doi.org/10.1089/omi.2011.0118)
31. Carlson, M.: org.Hs.eg.db: Genome Wide Annotation for Human. (2024). R package version 3.20.0
32. Yu, G.: Enrichplot: Visualization of Functional Enrichment Result. (2024). doi:[10.18129/B9.bioc.enrichplot](https://doi.org/10.18129/B9.bioc.enrichplot). R package version 1.26.3. <https://bioconductor.org/packages/enrichplot>
33. Davis, T.L.: Optparse: Command Line Option Parser. (2024). R package version 1.7.5. <https://CRAN.R-project.org/package=optparse>
34. Wickham, H.: Ggplot2: Elegant Graphics for Data Analysis. Springer, ??? (2016). <https://ggplot2.tidyverse.org>
35. McKinney, W., *et al.*: pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing* **14**(9), 1–9 (2011)
36. Chandra, R.V., Varanasi, B.S.: Python Requests Essentials. Packt Publishing Birmingham, UK, ??? (2015)
37. Hagberg, A., Conway, D.: Networkx: Network analysis with python. URL: <https://networkx.github.io> (2020)
38. Ari, N., Ustazhanov, M.: Matplotlib in python. In: 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), pp. 1–6 (2014). IEEE