

RESEARCH

# Mitocondrias anormales en el tejido muscular

Julia Del Río Toledo<sup>\*</sup>  
, Javier Méndez Parrilla  
, Juan Jesús Rus Muñoz  
and Benjamín Cortés Fernández

<sup>\*</sup>Correspondence:  
[juliadelrio2003@uma.es](mailto: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 [?]. 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 [?]. 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 [?, ?]. La disfunción mitocondrial contribuye a diversas patologías [?], neurodegenerativas, cancerígenas [?] o musculoesqueléticas [?].

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 [?] o el OPA1 (Optic Atrophy 1) [?]. Además, el ejercicio regular desencadena adaptaciones moleculares que optimizan el rendimiento muscular y la salud mitocondrial [?, ?]. Sin embargo, con el envejecimiento, la dinámica mitocondrial y la capacidad de mitofagia pueden verse comprometidas, afectando así la funcionalidad muscular [?, ?].

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. [?] La disfunción mitocondrial [?] en estos tejidos se ha asociado con varias patologías relacionadas con la edad, incluidas la insuficiencia cardíaca y la sarcopenia [?]. 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. [?]

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 [?].

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 [?].

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 [?].

**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 [?].

#### 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.[?]

**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.[?]

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.[?]

### 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[?] [?] [?].

**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. [?] [?] [?] [?].

**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. [?]

**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. [?]

**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. [?]

**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. [?]

*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. [?]

**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. [?]

**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. [?]

**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. [?]

## 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, en formato JSON (estándar ampliamente utilizado para el intercambio de datos estructurados en aplicaciones web), contenía una lista dentro del campo "genes", que incluía los nombres de los genes asociados al fenotipo. Este formato estructurado fue procesado utilizando la biblioteca *pandas*, lo que permitió organizar la información en un DataFrame, una estructura de datos tabular que facilita su análisis y manipulación.

Una vez organizados, los datos se depuraron para eliminar duplicados o entradas incompletas, con lo que se garantizó que la lista de genes obtenida estuviera lista para su uso en análisis posteriores. Finalmente, los genes se almacenaron en un archivo de texto estructurado para facilitar su acceso y uso en las etapas siguientes del estudio.

La lista de genes recopilada fue utilizada como entrada en un segundo script de Python, *fetch\_interactions.py*, que se encargó de analizar las interacciones proteicas entre los genes utilizando la API de **STRINGdb**. 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.

Utilizando dicha API, el script envió solicitudes POST que incluían los nombres de los genes formateados con %0d como delimitador, cumpliendo con los requisitos específicos de la API. Se configuraron parámetros como la especie de interés (*Homo sapiens*, código NCBI: 9606) y un umbral de confianza de 0.5, significando esto que las interacciones incluidas tienen al menos un 50% de probabilidad de ser funcionalmente relevantes, según los cálculos de STRINGdb.

La respuesta de la API, en formato JSON, proporcionó información detallada sobre las interacciones, incluyendo los genes conectados, la naturaleza de las interacciones y una puntuación de confianza. Además, el script incluyó una función para descargar y guardar una representación visual de la red directamente desde STRINGdb en formato PNG, lo que facilitó la documentación gráfica de las interacciones proteicas.

### 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 explorar relaciones clave entre los genes asociados a nuestro fenotipo. Para comenzar, los datos generados previamente con STRINGdb, que contenían la lista de nodos y aristas, fueron exportados en formato CSV y posteriormente importados en R mediante las funciones estándar *read.csv()*. Este enfoque aseguró la interoperabilidad entre Python y R, facilitando la integración del flujo de trabajo.

En R, los datos se transformaron en un grafo dirigido utilizando la función *graph\_from\_data\_frame()*. Este grafo representaba los genes como nodos y las interacciones entre ellos como aristas ponderadas, donde el peso correspondía a la puntuación de confianza calculada por STRINGdb. La dirección de aristas permitió representar el flujo funcional entre los genes.

La visualización inicial del grafo se realizó utilizando la función *plot()* de iGraph, con personalizaciones que resaltaban características clave de la red. Se ajustaron colores y tamaños de los nodos según su grado de conectividad, lo que permitió identificar rápidamente los genes más conectados o "hubs". Para optimizar la interpretación visual, se probaron diferentes algoritmos de disposición, como *layout\_with\_fr* (disposición de Fruchterman-Reingold) y *layout\_nicely*, que distribuyen los nodos de manera equilibrada según sus conexiones. Estas configuraciones facilitaron la identificación de patrones estructurales en la red y permitieron una representación gráfica intuitiva de las interacciones proteicas.

Además del grado de los nodos, se calcularon otras métricas topológicas relevantes, como la centralidad de intermediación, que identifica genes clave para el flujo de información, y el coeficiente de agrupamiento, que mide la tendencia de los genes a formar clústeres densos.

La distribución del grado de los nodos se realizó mediante histogramas, que mostraban cómo se distribuían las conexiones entre los genes. Adicionalmente, se generaron mapas de calor que representaban la intensidad de las interacciones (pesos de las aristas), proporcionando una visión global de las relaciones más relevantes. Estas visualizaciones se realizaron utilizando bibliotecas de R como *ggplot2* y *heatmap()*, que facilitaron la interpretación de los datos.

Finalmente, se evaluaron las comunidades detectadas mediante algoritmos de agrupamiento. Cada comunidad fue analizada en términos de su relevancia funcional en el contexto del fenotipo, considerando la conectividad interna y los genes principales en cada módulo. Este enfoque permitió identificar subgrupos funcionales de genes, lo que ayudó a interpretar mejor las relaciones biológicas subyacentes.

### 2.2.3 Análisis de enriquecimiento funcional

El análisis de enriquecimiento funcional se realizó en R utilizando el script *functional\_analysis.R*, diseñado para identificar procesos biológicos sobrerrepresentados entre los genes obtenidos en los pasos previos. Este análisis se enfocó exclusivamente en términos de procesos biológicos (GO-BP) utilizando el paquete *clusterProfiler* y la base de datos de anotaciones *org.Hs.eg.db*, que ofrece información detallada sobre genes humanos.

El script procesó archivos de texto que contenían listas de genes, organizados en un directorio de entrada especificado por el usuario. Para cada archivo, los genes se leyeron y limpiaron utilizando funciones de R, asegurando que los identificadores fueran válidos para el análisis. A partir de estos datos, se aplicó la función *enrichGO()* de *clusterProfiler*, configurada para trabajar con símbolos de genes humanos (SYMBOL) y para identificar procesos biológicos relevantes. Se empleó el método de ajuste del p-valor de Benjamini-Hochberg (BH) para controlar la tasa de falsos positivos, con un umbral de significancia de 0.05, asegurando la robustez de los resultados obtenidos.

Cada conjunto de genes fue evaluado por su relación con términos de la ontología de procesos biológicos, generando resultados que incluyeron términos clave relacionados con el metabolismo energético y la regulación de procesos celulares. Para facilitar la interpretación, se generaron gráficos de puntos (dotplot) que representaban visualmente los términos enriquecidos junto con sus niveles de significancia. Estos gráficos se guardaron como archivos PNG en un directorio de salida definido por el usuario, garantizando una documentación clara y reproducible de los resultados.

Con este script se automatizó el análisis para múltiples conjuntos de genes, procesando todos los archivos presentes en el directorio de entrada. Esto permitió una ejecución eficiente, reduciendo la intervención manual y asegurando la consistencia en el procesamiento de los datos.

### 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.

### 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.